CHARLES STARK DRAPER LAB INC CAMBRIDGE MA

JOVIAL STRUCTURED DESIGN DIAGRAMMER (JSDD). VOLUME III. PROGRAM--ETC(U)

F30602-76-C-0408 AD-A052 733 UNCLASSIFIED R-1120-VOL-3-PT-3 RADC-TR-78-9-VOL-3-PT-3 OF 3 AD AO52733 Marie Marie

හ DA 05278

RADC-TR-78-9, Vol III, Part 3 (of four) Final Technical Report February 1978



JOVIAL STRUCTURED DESIGN DIAGRAMMER (JSDD), Value IIT Program Description, P. + 3

- G. Goddard M. Whitworth
- E. Strovink

The Charles Stark Draper Laboratory, Inc.

UC FILE COPY

DDQ ROBUM IN APR 17 1978 SUMIV 5

Approved for public release; distribution unlimited.

## COPY AVAILABLE TO DDG DOTS NOT PERMIT FULLY LEGIBLE PRODUCTION

ROME AIR DEVELOPMENT CENTER Air Force Systems Command Griffice Air Force Bose, New York 13441 Because of the size of this volume, it has been divided into four parts.

Part 1 contains pages 1/2 - 123, 649 - 657, Part 2 contains pages 124 - 344.

Part 3 contains pages 345 - 592, Part 4 contains pages 593 - 648.

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-78-9, Vol III, Part 3 has been reviewed and is approved for publication.

APPROVED:

Sonald Van alatine

DONALD VANALSTINE Project Engineer

APPROVED:

Mendal Co Souman

WENDALL C. BAUMAN, Colonel, USAF Chief, Information Sciences Division

FOR THE COMMANDER:

JOHN P. HUSS

Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (ISDA) Griffies AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

18) RADC (19) TR-78-9-VOL-3-PT-3

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered) READ INSTRUCTIONS BEFORE COMPLETING FORM REPORT DOCUMENTATION PAGE 2. GOVT ACCESSION NO. 3. RECIPIENT'S CATALOG NUMBER RADC-TR-78-9, Vol III, Part 3 (of four) TYPE OF BESCHT & PERIOD COVERS IOVIAL STRUCTURED DESIGN DIAGRAMMER (JSDD) Final Pechnical Reports Jolume III. Program Description September 76 - October 77 G. Goddards F3Ø6Ø2-76-C-Ø4Ø8 M. Whitworth Strovink GANIZATION NAME AND ADDRESS The Charles Stark Draper Laboratory, Inc. 555 Technology Square P.E. 62702C J.O. 55811412 Cambridge MA 02139 11. CONTROLLING OFFICE NAME AND ADDRESS Feb. Rome Air Development Center (ISIM) Griffiss AFB NY 13441 248 14. MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office) 15. SECURITY UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING 16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same 18. SUPPLEMENTARY NOTES RADC Project Engineer: Donald VanAlstine (ISIM) 19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Structured programming Preprocessor Structured design diagram Flowcharter Structured extension JOVIAL J3 Parse Invocation diagram Parser generator 20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The report presents a detailed description of the JOVIAL Structured Design Diagrammer program implementation for purposes of maintaining or modifying the system. DD 1 JAN 73 1473

EDITION OF I NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

grant grant and a series grant and a series (complete) and the contract separate a finished the second second second second FIATERTS 10 Asserted for guidle release; discriberton unilmited. owe.3 Assert the control of the control of the control of the Color Structured Decision of the Color Structured Decision of the Color Structured Decision of the Color of the Color

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Date Entered)

Phase 2 Structured Design and Invocation Diagrams

ACCESS 114 NTIS		on #
DDC	3 . Se	<u></u>
NA****	0.0	0
18 '''	1-1	
	erica hanah keri	23023 V
ev <b>85</b> 78 <b>22</b> 11 5,	OK/AVNI AST S	Y COMES

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER

THIS LISTING CONSISTS OF OUTPUT FROM THE CHARLES STARK DRAPER LABORATORY'S JOVIAL J3 STRUCTURED DESIGN DIAGRAMMER.

PRINCIPAL DESIGNERS AND IMPLEMENTORS

GARY W. GODDARD, CSDL STAFF
HARK H. WHITWORTH, CSDL STAFF
ERIC F. STROYINK, GRADUATE STUDENT, M.I.T.
COMPUTER SCIENCE DIVISION
THE CHARLES STARK DRAPER LABORATORY, INC., CAMBRIDGE, MA.

346

Phase 2 Structured Design and invocation Diagrams

٥.
в
н
п
STATES
и
1
0
м
100

CENTER NOT THE TENT NOT THE TEN

TABLE OF CONTENTS	CHVERT 179	SPACES 184	NULL 185	
TABLE OF CONTENTS	ERT	CES	T	GTM

. 0	<b>3</b> .4	S	ė.
-	-	-	
			60
11.5			
D23 V.S			
981			90.
	200		
100			
			-
			011
100			
125			
3710-2			70.0
2.5	34	8.	E.
			B.
1000			8
	•	•	
			2/
	•		
2			
			10
1.000			95
B. 1889			86
Ballie I	38.5		ile:
UERT 179	ACES 184	185	TAKE TAKE
D 12 14	S.		Œ.
2	w	B	8.0

```
* **CONTROL*1 THRU CONTROL*4 ARE * STHT*TYPE CONSTANTS FOR **
                                                                                                                                                                                                                                                                                                                                                                                                                                                         *ITEH CUR*REC INTEGER P 0 $ *** CURRENT FILE 3 RECORD ** *** SITE DELIM*COMMENT INTEGER P *** ** SIMITIVE CONSTANT ***
                                                                                                                                                                                                                                                                                                                                                                  *************************
                                                                                                                              *ITEH CONTROL*4 INTESER P 11 8
***ORIF CLAUSE OR INSTANCE **
***ITEN CUR*GROUP INTEGER P 0 8
***INDEX INTO GROUP OF CURRENT
**TREE**
                                                  *ITEM CONTROL*1 INTEGER P 8 $
***PROGRAM, PROC OR CLOSE
* MEAD**
       START 8
**********
```

*ITEM OLSP INTEGER \$ *** PTZ DISPLICEMENT FOR ** RELATIVE LINE NUMBERING** ** IEM OLSPLAY*LINES INTEGER P ** S	INTEGE PLACEMEI LINE NI AY'LINE	S IN FO	ING.		3.
***LINES SPANNED BY A STUMP R. * BOX** * ITEM DISPLAY**WIDTH INTEGER P	AY*WID	4 %	STUMP	P 6.	
TIEN END'SCOPE INTEGER P.	END'SCOPE INTEGER PAT'TYPE CONSTANT	REF	80X	• • • •	
FILE BOFILE B P 0 S  FILE F B C F F F F F F F F F F F F F F F F F	NO OF	ILE B	FLAG.		
BLOCK				•	

0 press (10)	N S ZIS	1 AP				*******
	TABLE FI'BUF R FI'BLKSIZ S N S	INTEGER	INTEGER	INTEGER	INTEGER	******************
181	F1-80F	FIELDI				******
	TABLE		H311.			

	TABLE F2-BUF R F2-BLKSIZ S N &		*ITEH F3*AVAIL INTEGER P 1 8 ** *** THE NEXT EMPTY FILE 3 RECORD*
	2		
N	: 0		TE TE
W	12	-	a. m
1	3	12	ER
₩		. AC	FI
=	2	3	25
15		ō	75
F2-BUF IS THE FILE 2	100	*BEGIN *ITEM FZ*LINE CHARACTER \$ *END	A
88		7	N X
12		2	MZ
** F2*8UF BUFFER**	E.	A.	= ¥
2200	13	*ITE	=::
1000			

. .. F. B. OUF 1S THE FILE 3 BUFFER

TABLE F3.BUF R F3.BLKSIZ S N 8°  BEGIN  ITEM F4.END INTEGER 8  ITEM STAT.OUL INTEGER 8  ITEM WPTR INTEGER 8  ITEM WPTR INTEGER 8  ITEM BACK.V INTEGER 8  ITEM STAT.OUT STATE FILE 4 BUFFER				
BEGIN BEGIN BEGIN BEGIN BEGIN BEGIN BEGIN BEGIN BITEM F4.86GIN INTEGER 8 IITEM STATT-COL INTEGER 8 IITEM STATT-COL INTEGER 8 IITEM STATT-COL INTEGER 8 IITEM STOP-COL INTEGER 8 IITEM STOP-COL INTEGER 8 IITEM HOPT INTEGER 8 IITEM HOPT INTEGER 8 IITEM HOPT INTEGER 8 IITEM VPTR INTEGER 8 IITEM VPTR INTEGER 8 IITEM VPTR INTEGER 8 IITEM NPTR INTEGER				
BEGIN INTEGER 8 ITEM F4.8EGIN INTEGER 8 ITEM STATTCOL INTEGER 8 ITEM HPTR INTEGER 8 ITEM WOPTR INTEGER 8 ITEM WOTR INTEGER 8 ITEM WORK V				
BEGIN  BEGIN  ITEM F4.8EGIN INTEGER 8  ITEM STAFT-COL INTEGER 8  ITEM STAFT-COL INTEGER 8  ITEM STAFT-COL INTEGER 8  ITEM STOP-COL INTEGER 8  ITEM STOP-COL INTEGER 8  ITEM STOP-COL INTEGER 8  ITEM STOP-COL INTEGER 8  ITEM STOP-CINEGER 8  ITEM STOP-CINEGER 8  ITEM STOP-CINEGER 8  ITEM N-PTR INTEGER 8  ITEM N	*************			
BEGIN  ITEM F4.8EGIN INTEGER 8  ITEM STMT.UNIT INTEGER 8  ITEM STMT.UNIT INTEGER 8  ITEM BLOCK-WICH INTEGER 8  ITEM MIDPT INTEGER 8  ITEM W.PTR INTEGER 8  ITEM W.PTR INTEGER 8  ITEM W.PTR INTEGER 8  ITEM MACK-H I	BLE F3. BUF	3.8	\$ 21	
BEGIN INTEGER 8  ITEM F4.00 INTEGER 8  ITEM STAT.ONI INTEGER 8  ITEM STAT.ONI INTEGER 8  ITEM HODY INTEGER 8  ITEM HODY INTEGER 8  ITEM STAT.COL INTEGER 8  ITEM STAT.COL INTEGER 8  ITEM STAT.COL INTEGER 8  ITEM STOP.COL INTEGER 8  ITEM POTR INTEGER 8  ITEM WOTR INTEGER 8  ITEM BACK.V INTEGER 8  ITEM WOTR INTEGER 8  ITEM WO				•
ITEM F4.86GIN INTEGER 8  ITEM STMT.UNIT INTEGER 8  ITEM STMT.UNIT INTEGER 8  ITEM BLOCK.WIOTH INTEGER 8  ITEM BLOCK.WIOTH INTEGER 8  ITEM STOP.COL INTEGER 8  ITEM STOP.CLINE INTEGER 8  ITEM STOP.LINE INTEGER 8  ITEM STOP.LINE INTEGER 8  ITEM WPTR INTEGER 8  ITE	EGI			
ITEM STATTON INTEGER S ITEM STATTON INTEGER S ITEM STATTON INTEGER S ITEM BLOCK-WIDTH INTEGER S ITEM MIDPT INTEGER S ITEM MIDPT INTEGER S ITEM STOP-COL INTEGER S ITEM HOPT INTEGER S ITEM HOPTR INTEGER S ITEM HOPTR INTEGER S ITEM HOPTR INTEGER S ITEM HOPTR INTEGER S ITEM BACK-H INTEGER	TEM FL. BEG	NTEGE		
ITEM STMT UNIT INTEGER 8  ITEM 8LOCK UNITEGER 8  ITEM 8LOCK WIDTH INTEGER 8  ITEM STOP COL INTEGER 8  ITEM STOP COL INTEGER 8  ITEM STOP LINE INTEGER 8  ITEM LINES INTEGER 8  ITEM HPTR INTEGER 8  ITEM WPTR INTEGER 8  ITEM BACK V	FL.END	EGER	100	•
ITEM START-COL INTEGER 8 ITEM HIDTH INTEGER 8 ITEM HIDTH INTEGER 8 ITEM STOP-COL INTEGER 8 ITEM STAFF-LINE INTEGER 8 ITEM STAFF-LINE INTEGER 8 ITEM STOP-LINE INTEGER 8 ITEM WOPTR INTEGER 8 ITEM WOPTR INTEGER 8 ITEM WOPTR INTEGER 8 ITEM WOTR INTEGER 8 ITEM WOTR INTEGER 8 ITEM WOTR INTEGER 8 ITEM WORTH INTEGER 8 ITEM WOTR INTEGER 8 ITEM WORTH IN	H STHT .U!	INTEG		•
ITEM BLOCK-WIDTH INTEGER 8  ITEM STOP-COL INTEGER 8  ITEM STARY-LINE INTEGER 8  ITEM STOP-LINE INTEGER 8  ITEM H-PTR INTEGER 8  ITEM W-PTR INTEGER 8  ITEM	H START	INTEG		
ITEM MIDPT INTEGER 8  ITEM STOP-COL INTEGER 8  ITEM START-LINE INTEGER 8  ITEM LINES INTEGER 8  ITEM H-PTR INTEGER 8  ITEM H-PTR INTEGER 8  ITEM BACK-H INTEGER	H BLOCK.	THI H	3	
ITEM STOP-COL INTEGER 8 ITEM LINES INTEGER 8 ITEM LINES INTEGER 8 ITEM H-PTR INTEGER 8 ITEM W-PTR INTEGER 8 ITEM W	H MIDPT	GER S		
ITEM START-LINE INTEGER 8 ITEM LINES INTEGER 8 ITEM H-PTR INTEGER 8 ITEM W-PTR INTEGER 8 ITEM BACK-H INTEGER 8 ITEM BACK-V INTEGER 8 END	M STOP-CC	NTEGE		•
ITEM LINES INTEGER 8 ITEM NOT RINEGER 8 ITEM W'PTR INTEGER 8 ITEM W'PTR INTEGER 8 ITEM BACK'H INTEGER 8 ENO	H START"	INTE	W	
ITEM STOP-LINE INTEGER 8 ITEM H-PTR INTEGER 8 ITEM BACK'H INTEGER 8 ITEM BACK'V INTEGER 8 END END "F4"BUF IS THE FILE 4 BUF	H LINES	GER S		
ITEM H-PTR INTEGER 8 ITEM V-PTR INTEGER 8 ITEM BACK-W INTEGER 8 ENO	TEM STOP-LI	INTEG	~	
ITEM V'PTR INTEGER S ITEM BACK'H INTEGER S ENO	TEN H-PTR	GER		
ITEM BACK'H INTEGER \$ ENO	ITEM V.PTR	GER		•
END ENDf. BUF IS THE FILE 4 BUF	ITEN BACK'H	393		
END FF. BUF IS THE FILE & BUF	ITEM BACK'V	EGE	THE PERSON	
4-BUF IS THE FILE 4 BUF	END			•
.4.8UF IS THE FILE 4 BUF	•			:
.4.8UF IS THE FILE 4 BUF	AND STREET			
.4.BUF IS THE FILE 4 BUF	STATES AND SE			
	4.80F			FFER
	A PONT TOWN			
	. LEN RESPONDEN			
	The state of the s			

\* TABLE F4\*BUF R F4\*BLKSIZ S N S\*
\* TABLE F4\*BUF R F4\*BLKSIZ S N \*
\* SECONT TO THE SEC OVERLAY LINES OUT = F2 REC S POVERLAY MAX-LINE LNGTH = POPEND PROPERTY NO PERSON PROPERTY PRO . FIELDS.

. .. GROUP IS THE FILE 3 TREE.

PESIGN DIAGRAM OF THE DESIGN DIAGRAMMEN

"ITEM FRON'PAGE INTEGER \$

"ITEM FRON'PAGE INTEGER \$

"ITEM PAGE"RE INTEGER \$

"ITEM F2" BLK INTEGER \$

"ITEM F2" BLK INTEGER \$

"ITEM F4" BLK INTEGER \$

""" NUMBER OF FILE 4 BLOCK IN

""" F1 EFILE 7 BLOOD V 309 16

""" F1 EFILE 7 BLOOD R 316 15

"" F1 EFILE 7 BLOOD R 316 15

""" F1 EFILE 7 BLOOD R 316

. \*\*FILE3\*INCLUSION IS A FLAG \* INDICATING WHETHER A FILE 3

```
C S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRAMER
Design Diagram of the Design Diagrahmer
                                            REC SHOULD .. BE CREATED FOR THE CURRENT FILE 1 REC (PT1)."
                                                                                                                      **********************
                                                                                                                                                                                                                                                                                               GROUP ..
                                                                                                                                                                                                                                                                                                                               "ITEM GROUP" MAX INTEGER P 499 S"
                                                                                                                                                              *ARRAY GROUP-STACK 100 INTEGER
                                                                                                                                   *FILE FINAL"OUT H 20000 V 150
                                                                                                                                                                                                                                                                      *ITEM GROUF AVAIL INTEGER P 1
                                                                                                                                                                                                                                                                                                                                                              *** THE SIZE OF GROUP **
```

\*\*\* "FEM LAST\*, INTEGER P - 1 \$

\*\*\* "PT1 INDEX OF LAST FILE, REC \*\*\* OF A SET\*\* "\*\* "PT1 INDEX OF LAS\*\*
\*\*\* "FEM LAST\* FI INTEGER P - 1 \$

\*\*\* "PT1 INDEX OF LAST FILE1 REC \*\*\*
\*\*\* "PEAD" \*ITEM M\*FATHER INTEGER \$ \*ITEM H"SPACE" INTEGER P 4 8 \*\* HORIZONTAL SPACING CONSTANT\* \* \*\*\* PT2 STORES H" PTR OF CUR'REC \*ITEM H\*SPACE"Z INTEGER P 2 8
\*\*\* HOFIZONIAL SPACEING
\* CONSTANT \*\* \*ITEH HEADROOM INTEGER P & S \*\*\*LINES SPANNED BY PAGE \* HEADING\*\* \* HISTORY\*\*

• ••LAST\*LINE IS USED IN PART 1.
• TO STORE THE LAST RELATIVE ••
••• LINE NUHBER IN THE CURRENT
• TREE, IN PART2 IT IS THE
• ABSOLUTE •
• •• LAST LINE WHICH HAS BEEN

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF THE DESIGN DIAGRAMMER

OUTPUT ..

TIEM LAST PROC INTEGER \$

"ITEM LAST" PROC INTEGER \$

"ROC"

"ITEM LAST" STUMP INTEGER P 0 8

"ITEM LAST" STUMP OF THE LAST \$

STUMP STACK 100 INTEGER \$

"STUMP STACK 100 INTEGER \$

"STORES HISORY OF LAYOUT STACK TOP INTEGER \$

"TEM LAYOUT STACK TOP OF CURRECT STACK TOP OF CURRECT TOP OF CURRECT STACK STORES HIDPOINT INTEGER \$

"STORES HIDPOINT INTEGER \$

"ST

""NAME"BYTE IS THE BYTE OF HEADER AT WHICH THE MODULE NAME PRINTS"

TITCH NAME OF TE INTEGER 8

TITCH NEWFILES B P 0 8

TITCH NEWFILE B P 0 8

\*\* FITEM NEW\*TEXT CHARACTER \$

\*\*\* PTS STORES TEXT EXTRACTED \*\*

\*\* FROM FILE2\*\*

\*\* TEM NULL\*SCOPE B P B \$

\*\*\* PTS FLAGS CONTROL PHRASE\*S \*\*

\*\* NULL SCOPE\*\*

\*\* TEM JOJU BLOCK IN CORE\*\*

\*\*\* PTS PUTOUT BLOCK IN CORE\*\*

"OUT'LINE IS PUTOUT'S BUFFER

CHARAY OUT-LINE OUT-BUF'SIZE CHARAY OUT-LINE OUT-BUF'SIZE CHARACE BYTE INTEGER 8

"HEADER BYTE AT MICH PAGE'NO!"
"ITEM PAGE'NO INTEGER 8

"PTEM PAGE'NO INTEGER 8

"PTEM PAGE'NO OUT-BURNER PAGE 6

"TEM PAST'NIO 8 8

TEMPORARY TOP OF PROC'STACK\*\*\*
TIEM READ-ONLY SMITCH FOR TILE\*\*
TIEM PROC'COUNT INTEGER P 0 8 \*\*\* THE NUMBER OF PROCS IN THE \*\*\* PROGRAM \*\*\*
FILE PUTOUT'S H 20000 V 150 16\*\*

C. S. DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF THE DESIGN DIAGRAMMER

		• •		•	• •	•	•		•	•	٠.	•	•	•	•	• •		•	• •		•	•	• •	•	•			•	•	• :			•	:			• •	•	•
11.6	•		۵					,									CUR' REC"			- September 1				-		Y.	•			-	2			CUR. REC	ENP	:		:	
IN THE FILE	NTEGER	FILES	NTEGER	FILES		9										9	8.8		INTEGER \$	1	*		<b>~</b> :			TUMP	0				CHARACIERS	1		3	:	GES	002	RSAL.	GER
=	INT		INT	E N	FEFE	COL		DOTATION							100		. 5	7	EG	-	GER	7	EC	2	•	A SI	373	ENTRY		ER .	¥ .	·	ER		ER		*	RAVE	NTE
=		2		-	I		J	2 0		•	TA	BOUR		6 30	Š	959	INE OF			^	INTE	UNIT	Z.	3		L.	III S	4		2	-	SE	w		ACT	MES	TAC	-	
R OF RECS	BLK3	RECS	.IN. BLK4	RECS		ST		200	, ,	EGER P 7	ONO			. SPAC	2		I	Ť.	EN	5			NOTH		3	300 V	2004	GROUP		<b>∢</b> ι		·z	-	7	HAR	BUG ME		TREE	-
4	IN.		Z	9	T GHT COL	ES				Z	w 0			E. S	ET	۵ م د		27	0		TYP	HENT	-		. F.		٠.	·			2 2			2	5	96	VERS	~	RSE
¥ .	RECS.	R 0F	S	~	. 3	TORE	0	5 6	5 0		7	IN		NGL	S	N .	Z						I,	7	TUMP	-	1	3	HEA	귤 :	TEMPT			STA	MES	8	2 0	-	RAVE
LOCK		MBE		MBE	OCK.	2 5	. 86	¥ ,	NTE	SE	= 3	,	S	SI	HAN	BLE	2 5	Z		Į:	S	STA	ST	7 7	S		N N	, "	-			141		2	-	1000	FG	AC	-
8 6	W	NO.	TEN	₹.	150		3	200	POT	TEM	. 21	5 4	SPA	TEH	3	200	14	5.7	Ŧ.				TEN	STR	TEN	4	TEN	-	STU	TEN	1		TEM	. 61	-	USED	I NA		TEM
٠.		: .		:	. :	:	. :			:		:		:	:	. :	.:		::		:	:				: .	:	:		-	:	:	:	: .			٠.	:	::

G S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design Diagram of the Design Diagrammer

		**************************************	
TEGER S CUR'REG"  ER S UR'REG"  UR'REG"  FOR CONSTANT  FOR CURRENT  SWITCH**  SMITCH**  SWITCH**	ROUTINE	OR CUR'REC EQGET F1.REC S	MESSAGE 9M(END PARI) S - *OUT(MESSAGE, MESS'SM) S -
TIEM V'FATHER INTEGER S  "ITEM V'SON INTEGER S  "ITEM V'SON INTEGER 9  "ITEM V'SPACE INTEGER P 3  "ITEM WICH INTEGER P 3  "ITEM WICH INTEGER P  "ITEM WITES B S  "ITEM MRITES	INITIALIZE S  INTIALIZE S  INTERPRETATION OF THE STATE STATE S  IF DE BUGI S  IF DE BU	DD UNTIL TEOFILE OR CURREC EQ	IF OCBUGIS

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANMER DESIGN DIAGRAM OF THE DESIGN DIAGRAMMER

F DE BUG1 8 ***********************************		MESSAGE = 10H(PARTZ DONE) \$ +
.1.		
	*IF DE BUG1 8 *	-*MESSAGE = 10H(LAST LINE ) 8 +
*OUT(MESSAGE, MESS*SM *MESSAGE 12H(START *OUT(MESSAGE, MESS*SN *********************************		#1*MESS = IFORMAI(LAST*LINE) & # *MESSAGE = CAT(MESSAGE, T*MESS) # * *
**************************************		** ** ** ** ** ** ** ** ** ** ** ** **
		*OUT(MESSAGE, MESS'SM) & *
	YTE'EN \$ *	A The Control of the
		*PHZERR(MESSAGE) \$

PROC ACESSI (REC'NO) 8 .

PROVIDES THE INTERFACE FOR FILE 1. ITS PARABETER IS AN ASSOLUTE RECORD NUMBER. ACCESSI READS THE APPROPRIATE BLOCK OF FILE 1 RECORDS INTO COR AND RELURNS THE INDEX INTO FILE 1'S BUFFER OF THE DESIRED RECORD.  FILLI BLOCK IN TEGER 8 FECORD.  FILLI BLOCK IN HICH RECORD.  RESIDES ***  FILLI BLOCK IN HICH RECORD.  FILLI BLOCK IN BLOCK
---

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of access1

**************************************	
ALF DEBUG S	ACE SSI = 1 8 +
	'IF ACCESSI GQ RECS'IN'BLK1 OR " FILE1 EQ V(EOF1) \$

RETURN S

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of accesss

\*PROC ACCESS (REC\*NO) 8 \*

```
THE FOLOR NUMBER AND MAKES SUME
THAT THE FILE BLOCK
THAT THE FILE BLOCK
THAT THE FILE BLOCK
THAT THE FILE BLOCK
THE SAGE
THE FILE BLOCK
THE BLOCK
THE FILE BLOCK
THE FILE BLOCK
THE FILE BLOCK
THE B
```

	******	FOUT (MESSAGE, MESS'SW) &	
erf ACCESSZ GQ RECS'IN'BLK2 8 *****ACCESSZ # * 1 8 * esteretestestestestestestestestestestestestes	ACCESS2 = 1 & *		
*RETURN S .			

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design Diagram of Access3

\*PROC ACCESSIREC\*NO) \$ \*

THE FUNCTION ACCESS IS THE	
13 803 30	
* DADAMETED TO THE ABOUT HE	
PERSONAL TO THE ADSOLUTE	
RECORD NUMBER OF A FILE 3	
" KECORD. ACCESSS DETERMINES IF	
. THE BLOCK CONTAINING THE	
. RECORD IS IN CORE. IF IT IS.	
. THEN THE INDEX INTO THE	
. BUFFER OF THE RECORD IS	
. RETURNED. IF THE BLOCK IS	
• EXISTS BUT IS NOT IN CORE.	
THEN MOTTER TO EXAMINED. TE	
. TOTTES TO ONTO THE TO	
THE THE PROPERTY OF THE PARTY OF	
INEN, THE BUFFER HUST BE	
WRITTEN TO DISK (BY TRANSS EX	
* TRANSFER WRITE3). ACCESS3	
. READS THE DESIRED BLOCK INTO	
. THE BUFFER AND RETURNS THE	
. INDEX INTO THE BUFFER OF THE	
. RECORD. IF THE BLOCK DOES NOT	
* EXIST THEN ACCESS 3 RETURNS	
. THE INDEX INTO THE BUFFER OF	
THE RECORD AS IF THE RLOCK	
EXIST. AND SETS	
* RECS'IN BLKE TO THAT VALUE.	一年 中国
•	
***************************************	
STIFM ACCESSE TNIFGER &	
11 34	
* THE BECOOL .	
*** THE ABSOLUTE NUMBER OF THE	
* RECORD.	
*REMQUCIREC*NO. F3*BLKSIZ = *	
. BLOCK'NO, ACCESS3) \$	
***************************************	
Detail and the first	
OTE METERS AND F3-81K GO G AND *	
F3-BLK NG BLOCK-NG &	7 . de . d
	PERSON NETTE USI P.S. BLK
CONTRACTOR CONTRACTOR OF CONTRACTOR OF STREET	SEMULING BLUCK NO
C. S. Talester of either young to take by the state of the	SO SEEDINGS OFFICE STREET, TO SEED STREET, TO
	**************

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF ACCESS3

\*TRANSFER\* HRITES \$ \*

IF DEBUG4 8 TT MESSAGE = 6H(READ3 ) 8  **T MESS = IFORMAT(REC*N3) 8  **HESSAGE = CAT(MESSAGE, T*MESS) 8  **OUT(MESSAGE, MESS*SH) 8  **OUT(MESSAGE, MESSAGE,
---

*RECS'IN'BLK3 = ACCESS3  *RECS'IN'BLK3 = ACCESS3  **********************************

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMIER Design Diagram of Access&

THE FUNCTION ACCESS, IS THE SAME AS ACCESS, INTEGER S THE RECORD. THE		
S  WHE  WHE  WHOST HRITE OUT F4"BLK  BEFORE READING BLOCK"NO ".  TRANSFER" WRITE & S.  TRANSFER" WRITE & S.  TRANSFER" WRITE & S.  TRANSFER" WRITE & S.  THE DEBUGS S.  THE DEBUGS S.  THE SSAGE  WHE SSAGE		
BEFORE READING BLOCK"NO "  "TRANSFER" WRITE & "  "TRANSFER" WRITE	* THE FUNCTION ACCESS4 IS THE FILE 4 INTERFACE. IT IS THE SAME AS ACCESS3 EXCEPT THAT I DOESN*T CALL TRANSFER*HRITE4 IF READSH IS	
BEFORE READING BLOCK"NO "  "TRANSFER" WRITE & *  "TRANSFER" WRITE	NO .	
BEFORE READING BLOCK NO  TRANSFER WRITE & S  O AND  BLOCK NO EXISTS REA  IF DEBUGS  THESSAGE  THESSAGE  THESSAGE	ITEM ACCESS4 INFEGER \$  ITEM ACCESS4 INFEGER \$  ITEM ELOND    ITEM BLCCK NO INTEGER \$  ITHE FILE BLOCK CONTAINING*	
BEFORE READING BLOCK"NO ""  "TRANSFER" HRITE & S "  "TRANSFER & S "	THEM REC'NO INTEGER &  ""THE ABSOLUTE NUMBER OF THE *  "RECORO"*  "REMQUOIREC"NO, F4"BLKSIZ = *  "BLOCK"NO, ACCESS4) \$  "***********************************	
**************************************		WRITE OUT F4*BLK READING BLOCK*NO
**************************************	1 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1	NSFER-WRITEL S .
	* BLOCK NO LS F4 EMPTY \$	EXISTS REF

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of accesse

OORIN P.		FILE . FILE FILE FILE FILE
OPEN INPUT FILEY 2 = BLOCK'NG DOESN'T VET EXIST  'SHIT INPUT FILEY 2 = BLOCK'NG S'SHIT INPUT FILEY 3 = BLOCK'NG S'SHIT INPUT F	• •	
THAUL FILE, 2 BLOCK NO DOESN'T YET EXIST  SHALL INPUT FILE, 2 BCS' IN'BLK, 1 BLOCK NO STATE NOT	•	
POSITIES, 2 BLOCKNO DOESN'T YET EXIST  POSITIES, 1 BLOCKNO S  PLABUE  SHUT INPUT FILE, 2 BLOCKNO BLOCKS OR RECS'IN BLCK  SHUT INPUT FILE, 1 BLOCKNO BLOCKS OR RECS'IN BLCK  SHUT INPUT FILE, 1 BLOCKNO S  SHUT INPUT FIL	•	
SHOT INPUT FILE, 1 S	•	* SON (FILE 4.2) = BLOCK NO 8
*SMIT INDUFFLEY 1  *SMIT INDUFFL	• •	* FLET 2 RECS IN BLK++
THE		* SHUT INPUT FILE 4.2 8 *
IN FILE, IS  IN FILE, IS  ORIF F. Buk NO GLOCK-NO S  STATE IN THE FLECT IN BLK4 S  SHUT INDUT FILE, IS  SHUT INDUT FILE  SHUT IND	• •	*************************
IN FILE 1.  IN FIL	•	9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
TO PEN INDUITE TE TEST IN BLCK TO BE NITH FILE TO BE NITH THE TEST IN BLK TO BE NITH THE TEST IN BUT T		1 1
POEM INDUTFILE'1 " SOOR INDUTFIL		
POSEN INPUT FILE*1 8  **POSTILE*1 8	•	. IN FILE, 1
**OPEN INPUT FILE**1 & **OOFN INPUT FILE**1 &	•	
**************************************		
THE THE TIME T		*OPEN INPUT FILE4.1 \$
* FABUF & SHUT INDUT FILE* 1 &	•	+ FOCK FILE(+1) = 610K*NO S +
**************************************	•	TRACE & FECO. IN BERG.
**************************************		SHOT BILES. 1
TEITH ************************************	•	***************************************
**************************************	•	****************************
+-*ORIF 1. S*ORIF F.* GLOCK*NO S*ORIF 1. S *-*ORIF		**************************************
**************************************	•••	
+-ORIF F4-BLK NQ BLOCK'NO S+ ORIF F4-BLK NQ BLOCK'NO S+ ORIF F4-BLK NQ BLOCK'NO DOESN'T YE ORIF I S		
BLOCK*NO B **********************************		
BLOCK'NO DOESN'T YET E  "IFEITH "READSH & "	*****************	
FEITH ************************************	17010 N. 170-11 LTY0-11	•
PIETTH ***********************************	••	***** * **** * ***** * ***** * ****** *
IFELTH ************************************	•	
ORIF 1 &	••	************
ORIF 1	97 VA	
08.1F 1 \$		
	. · ·	1.
	Carry on the second control of	

 *IFEITH *	READSH S	* PIFEITH **+-*READSM \$ *4[F ACCESS4 GR RECS*IN°BLK4 \$ **ACCESS4 = - 1 \$ *	-*ACCESS4 = 1 S +
		+++RIFF 1 S +REGS'IN BLK4 = MAX(REGS'IN'BLK*	

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMIER Design diagram of access-out

THE INTERACE FOR THE PUTOUT THE INTERACE FOR THE PUTOUT OPERATION ARE ANALGOUS TO THOSE OF ACCESS. THOSE OF ACCESS. THOSE INTO LINESE \$ 1 THE BLOCK NO INTEGER \$ 1 THE BLOCK NO INTEGER \$ 1 THE RECORD IN PUTOUT REMANDIAGE OF THE \$ 1 THE RECORD IN PUTOUT REMANDIAGE OF THE \$ 1 THE RECORD IN PUTOUT REMANDIAGE OF THE \$ 1 THE BLOCK NO OUT SLK AND \$ 1 THE BLOCK NO OUT S	*PROC ACCESS OUT (REG"NO) S		
THE TO DO THE TENT OF THE TENT			
¥ #	THE INTERFACE FOR THE PUTDUIT FILES. ITS PURPOSE AND OPERATION ARE ANALAGOUS TO THOSE OF ACCESS		
F			
	*ITEM BLOCK*NO INTEGER \$ ***THE *BLOCK* CONTAINING THE * * RECORD**		
	*ITEM REC'NO INTEGER 8 *** ABSOLUTE NUMBER OF THE * * RECORD IN PUTOUT.*		
	*REMODOREC'NO, OUT'BUF'SIZE = * * BLOCK'NO, ACCESS'OUT'S *		
	"IF BLOCK"NO NO OUT" 9LK AND *		
**************************************		NEED TO WRITE OLD BLOCK BEFORE GETTING NEA ONE	A Transcription of the Control of th
**************************************	•		
		ASPEKTWRITE OUT M *	
	********* * BLOCK*NO LQ MAX*FOUTPUT	S SIJCK-NO EXISTS	139
* * BLOCK NO LG MAX FOUTPUT \$ ***********************************			A STATE OF THE STA
BLOCK NO LQ MAX FOULDUI S	•••		
BLOCK NO EQ MAX FOURDUR BLOCK NO EXISTS	•••	.!	HARRING HARRIST STATES OF THE SET 3 A
BLOCK NO LQ MAX FOUTPUT & BLOCK NO EXIS	•••••••••••••••••••••••••••••••••••••••		SS = IFORMAT (BLOCK NO) \$
BLOCK NO EXIS	••	* HESSA	IGE = CATINESSAGE, T. HESS)
BLOCK NO LQ MAX FOUTPUT & BLOCK NO EXIS		#JIno.	* OUTTIME COACE MESSING

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF ACCESS"OUT

CK DOESN" T EXIST	PUTOUT:2 PUTOUT:2  ** PUTOUT:2  ** PUTOUT:2  ** PUTOUT:2  ** POS(PUTOUT:2)  ** POS(PUTOUT:2)  ** PUTOUT:2  ** PUTOUT:3  ** PUTOUT:3  ** PUTOUT:3  ** PUTOUT:3  ** PUTOUT:4  ** PUTOUT:4  ** PUTOUT:4  ** PUTOUT:4  ** PUTOUT:4  ** PUTOUT:4  ** PUTOUT:5  ** PUTOU	FOR I = 0, 1, OUT BUF'SIZE - 1 * ********************************	**************************************	PUTOUT.1.**  **OPEN INPUT PUTOUT.1.*  **OUT********************************	FOR I = 0, 1, OUT BUF SIZE - 1 28 28	**************************************
	• • • • • • • •					**************************************

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of access out

. CORE ..

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF ACCESS"OUT

. 27 FROM 25 .

\*INPUT PUTOUT'2 OUT'LINE(SIS) S\*

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF ACCESS" DUT

28 FROM 25 \*

\*INPUT PUTOUT'1 OUT'LINE(SIS) S\*

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design Diagram of Advance-Page

PROC ADVANCE PAGE B

SE'S LINE'NO TO THE WUBER OF THE TOT THE WUBER OF EMTY PAGE. LINE'NO THE WUBER FROTTO PAGE NUMBERS TRON RELATIVE DNES.  TEM PAGE TATEGER OLRECTER OLSPECKENT OF LINE NUMBERS TRON RELATIVE DNES.  TEM PAGE TATEGER  CALLING REGALOT  ***ESSAGE = ICORNATILINE'NO; \$ *** ***ESSAGE = ICORNATILINE'NO; \$ *** **** ***** ***** **** ***** ***** ****		
STACEMENT USED SOLUTE LINE LLATIVE ONES.  LINE NO DN.  LINE NO DN.  SE LINE NO DN.  SE CATTINESS AND SE  OUTHESSAGE A CATTINESS. HES  SOUTHESSAGE A CATTINESSAGE.  PAGE LINE NO D S  "IT L	THE PROCEDURE ADVANCE PAGE	
THE NOTHEN SNACKEND THEN SNACKEND THEN SNACKEND TO SO OLUTE LINE LINE NO DN TENESS = 4H(ADV) & NESSAGE = IFORMAT(LINE'NO MESSAGE = AT(T'MESS, MES SNACKEND TO THESS HESS OUTHESSAGE MESS'SW) & THE SNACKEND TO	SO GOLDING OF CHOOSE STATE	
SOLUTE LINE SOLUTE SOLUTE LINE SOLUTE S	SEIS LINE NO 10 INE NUMBER OF	
SOLUTE LINE SOLUTE LINE LLATIVE ONES.  LINE NO DN  THESS E 4H(ADV)  THESS E 1FORMATICLINE NO  WESSAGE = IFORMATICLINE NO  WESSAGE = GAT(T MESS, MESS, MESS  OUTTON OF MESSAGE  THESSAGE = GAT(T MESS, MESS, MESSAGE  THESSAGE = GAT(T MESS, MESSAGE)  THESSAGE = GAT(T MESS, MESSAGE)  THESSAGE = GAT(T MESS, MESSAGE)  THESSAGE = GAT(T	THE TOP LINE OF THE NEXT	
SOLUTE LINE SOLUTE LINE LINE NO DN'  LUSED IN  STAGES = CATT HESS, HES  SOUTHESSAGE = LEORAT (LINE NO  HESSAGE = CATT HESS, HES  SOUTHESSAGE   LEORAT (LINE NO  HESSAGE = LINE NO	CHOCK TO STATE OF THE STATE OF	
SSTAGEMENT USED SOLUTE LINE LLATIVE ONES.  LLNE"NO ON"  STAGE IN THESS THANY IN THESS HESS  THESSAGE = IFORMATILINE"NO  THESSAGE = AND THESS HESS  THESSAGE = AND THESSAGE	202	
LINE NO DN°  LINE NO DN°  LINE NO DN°  SE USED IN  SAGE = IFORMAT(LINE NO  SAGE = CAT(T" MESS. MESS  SAGE = CAT(T" MESS. MESS. MESS  SAGE = CAT(T" MESS. MESS. MESS. MESS  SAGE = CAT(T" MESS. MESS. MESS. MESS. MESS  SAGE = CAT(T" MESS.	BECOMES THE DISPLACEMENT USED	
LIATIVE ONES.  LINE NO DN.  LINE NO DN.  LINE NO DN.  SAGE = LFORMAT(LINE NO PAGE SAGE = LFORMAT(LINE NO PAGE PAGE NO PAGE PAGE NO PAGE PAGE PAGE PAGE PAGE PAGE PAGE PAGE	FOR FINDING ABSOLUTE LINE	
LINE NO ON SED IN SSAGE = LORHAT (LINE NO PAGE OUT (MESSAGE = LORHAT (LINE NO PAGE LINE NO PAGE LINE NO OS SET		
LINE NO DN*  LUSED IN  THESS = LH(ADV ) S  HESSAGE = LATT HESS, HESS  OUTHESSAGE   LESS'SW) S  COUTHESSAGE   LINE NO PAGE  PAGE   LINE NO O S  TIL LINE NO O S		
USED IN  THESS = 4H(ADV) \$  HESSAGE = IFORMATILINE'NO  HESSAGE = ATORNATILINE'NO  HESSAGE = ATORNATILI		
LINE NO DN°  SUSED IN  FESSAGE = IFORMAT(LINE NO  FESSAGE = CAT(T MESS, MESS  COUT(MESSAGE, MESS'SW)  FESSAGE = LINE NO, PAGE  FERMUNO(LINE NO, PAGE  FACE LINE NO OS  FIF LIN		
LINE NO DN.  USED IN  WESSAGE = IFORMAT(LINE NO MESSAGE = CAT(T MESS, MESSAGE = CAT(T MESS, MESS OUT (MESSAGE, MESS SN) &  THE STATE OF THE NO PAGE		
S USED IN S AH(ADV) S HESSAGE A CAT(T'MESS, MESSAGE A CAT(T'MESS, MESSAGE A CAT(T'MESS, MESS, MESSAGE, MESS'SW) S TOUT (MESSAGE, MESSAGE, MESS'SW) S TOUT (MESSAGE, MESS'SW) S		
USED IN  T. MESS = LH(ADV) & MESSAGE = LATT HESS, HESS  OUTHESSAGE = CATT HESS, HESS  OUTHESSAGE   LINE NO, PAGE  PAGE, LINE S	FITEM LINE INTEGER \$ .	
USED IN  THESS = LH(ADV ) S  HESSAGE = LF ORMATICLINE'NO  HESSAGE = CAT(T'MESS, MESS'SM) S  OUT(MESSAGE, MESS'SM) S  PAGE, LINE'NO, PAGE  PAGE, LINE'S S  LIFLINE'NO OS  LI		
THESS = 4H(ADV) SHESSAGE = IFORMAT(LINE'NO MESSAGE = CAT(T'HESS, MESSAGE, HESS'SW) SHOUTHESSAGE, HESS'SW) SHESS'SW) SHESS'SW'SW'SW'SW'SW'SW'SW'SW'SW'SW'SW'SW'S	* CURPENT PAGE**	
TT HESS = 4H(ADV) & HESSAGE = IFORMAT(LINE NO PAGE SOUT(HESSAGE, HESS SW) & COUT(HESSAGE, HESSAGE, HES	FITEM PAGE INTEGED &	
THESS = 4H(ADV) S  HESSAGE = IFORHAT (LINE'NO PESSAGE = CAT(T'MESS, MESS POINT (LINE'NO PESSAGE, MESS'SW) S  THE SAGE = LINE'NO PAGE PAGE   LINE'NO PAGE   LINE'N PAGE   L	*** A DUMMY VARIAGE USED IN .	
THESS = 4H(ADV) \$ HESSAGE = IFORMATILINE'NO HESSAGE = GAT(I'ME'NO'NO'NO'NO'NO'NO'NO'NO'NO'NO'NO'NO'NO'	CALLING REMOUNT.	
## SSAGE # IF ORMAT (LINE NO PAGE # CATT HESS, HES ## CATT HESS, HES ## CATT HESS, HESS ## CATT		
HESSAGE = LFORMAT(LINE'NO MESSAGE = CAT(T'MESS, MESS, OUT(MESSAGE, MESS'SW) & CAT(T'MESS'SW) & CAT(T'MESS'SW) & CAT(T'MESS'SW) & CAT(T'MESS'SW) & CAT(T'MESS'SW) & CAT(T'ME'NO, PAGE, LINE'N & CAT(T'ME'NO, PAGE, LINE'N & CAT(T'ME'N) & CAT(T'M		
HESSAGE = LFORMAT (LINE'NO PAGE = LFORMAT (LINE'NO PAGE = CAT(T'MESS, MESS, ME		
THESS = 4H(ADV) \$ HESSAGE = IFORMAT(LINE'NO HESSAGE = GAT(T'MESS, MESS, MESS'SM) \$ OUT(MESSAGE, MESS'SM) \$ OUT(MESSAGE, MESS'SM) \$  OUT(MESSAGE, MESSAGE, MESS'SM) \$  OUT(MESSAGE, MESS'SM) \$  OUT(MESSAGE, MESS'SM) \$  OUT(MESSAGE, MESSAGE, MESS'SM)	***************************************	
HESS = 4H(ADV) \$ HESSAGE = IFORMAT(LINE'NO PESSAGE = CAT(T'MESS, MESS, MESS, MESS, MESS, MESSAGE, HESS'SW) \$ ************************************		
HESSAGE = IFORHAT (LINE 'NO 'S 'HESSAGE = CAT(THESS, HESS', HESS'		
## SSAGE # IF ORMAT (LINE "NO # SSAGE # CAT(T MESS, MES # OUT(MESSAGE, MESS'SN \$ # OUT(MESSAGE, MESS'SN \$ # PAGE, LINE NO, PAGE # PAGE, LINE NO OS *********************************	SS ILLESS	
*** COUTCHESSAGE ** CATCT MESS. MES ************************************		
**************************************		
**************************************		
PAGE, LINE, NO, PAGE, PA	*OUT (MESSAGE, MESS'SW) \$	
PAGE, LINE'S PAGE,	***********************	
PAGE, LINE S PAGE,		
PAGE, LINE) S		
S O C C X	O DACE I THE !	
TE LINE NO O S		
TE LINE NO 0 8		
IF LINE NO OS		
TILLING NO OR		************
	TE LINE NO O S	TACONIO + DAGE . I NOTH
Printed Sty. Expression of the contract of the	UNIT	THE NO THE CHOIN
OUISP = LINE'ND & **		
THE TOWN S .		
S ROLL S	00000	
	LINE NO S .	
	THE CORN	
	***************************************	

C. S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of Box\*hap

**************************************	
. "THE FUNCTION BOX"MAP ACCEPTS	
* STAT*TOKENS USED BY PHASE 1	
. TYPES USED BY PHASE2	
SECTION BOX MAP INTEGER S	
*ITEM STATTTYPE INTEGER 8 *	
essithe Input PARAHETERs.	
•	
orrecoss serventer EQ 1 8 sections 1 8 s	
•	
S S S S S S S S S S S S S S S S S S S	
	***************************************
**************************************	
	***************
The state of the s	
	1
	*******************
***************************************	*****************
HIEITH	SELIM \$ ********* = SEQ \$ *
	**************
	************************
ALL STREET, ST	
	****************
***************************************	
+-+04IF STHT*TYPE EQ 8 4*80X*HAP = PGH*TAIL 8 *	
d a city i delicated to the constitution of the city o	
4 710 - LEE XOD	

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of Box\*Map

-ORIF STHT TYPE LQ 41 8BOX*HAP = CONTROL*1 8BOX*HAP = CONTROL*2 8BOX*HAP = CONTROL*2 8BOX*HAP = CONTROL*2 8BOX*HAP = CONTROL*3 8BOX*HAP = CONTROL*3 8
---

TORST MATERIAL ON STATEMENT OF STATEMENT OF

FFR DISPERS OF CONTRACT SERVINGE PRIZED CONTRACT

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF BYTE'EM

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF BYTE"EM

W MR GIAR O TURNING THE STREET STREET	* & "*Suddes Tides aber *** # : 1390ses	elfelth +-++J EQ MAX*FOUTPUT \$ =+REMQUO(LAST*LINE, DUT*BUF*SIZE*  **********************************	ORIF 1 S	**************************************
* 33 FROH 32 *		THEFIT	ON I S	"IFEITH "J EQ OUT BLK S "

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANMER DESIGN DIAGRAM OF BYTE"EM

34 FROM 32 *	
Marie San Contract State of the party secure party secure	
FERTIT * EQ MAX-FOURDUR S.	INTERIM *-+JEQ MAXFOUTPUT \$ *REMOUCLASTALINE, OUTBUF'SIZE* ***********************************
MAIL AND	
IFEITH EQ OUT BLK S	******** *****************************
	+POS (PUTOUT*1) = J ** + OUT*8UF*21

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMIER DESIGN DIAGRAM OF CLOSE"REC

	- HAVE A CONTROL BOX	*IF STHT*TYPE GQ CONTROL*1 AND *	JNIT (SACCES	*ITEM TEMPIZ INTEGER \$	INTEGER	* BEING CLOSED**	*ITEM TERM*REC INTEGER S		NYROL BOX IT	
	CUTTO COO U		HAVE A CONTROL BOX	HAVE A CONTROL BOX	HAVE A CONTROL BOX	HAVE A CONTROL BOX	HAVE A CONTROL BOX	HAVE A CONTROL BOX	NIMAFR	
]	TOTAL STREET			IS RECORD	IS RECORD	S S S S S S S S S S S S S S S S S S S	S S S S S S S S S S S S S S S S S S S	LOSE REC S AND S AND S PLACE HE PART 1 FILE 3 FILE 3 FILE 3 FILE 6 FILE 6 FILE 6 FILE 7 FILE		
HAVE A CONTROL BOX HUST HAVE AN OOD NUMBER				JULT (SACCES*  11S RECORD  FEM*REC'S)*	S.  UNIT(\$ACCES*  IS RECORD  IF RECORD  FERM*REC(S)*	S S S JULT (SACGES* IIS RECORD	S S S S S S S S S S S S S S S S S S S	LOSE REC S AND S AND S PLACE THE PART 1 FILE 3 FILE		a SIMI aTYPE 8
		TO THE STATE OF TH	**SITERM REC)E) /) \$  ***********************************	*SIMITYPE =(. SIMITUALITIES ** *SIMITYPE = (. SIMITUALITIES ** *SITEMPRECIE) /) & *SITEMPRECIE // SIMITUALITIES ** ** ** ** ** ** ** ** ** ** ** ** **	**************************************	***TEM TEMPI INFEGER **  ***TEM PORARY INFEGER **  ***TEM PEMPIZ INFEGER **  ***TEM POWARY INFEGER **  ***TEM PEMPIZ INFEGER **  **TEM PEMPIZ INFEGER **  ***TEM PEMPIZ INFEGE	**RECORO NUMBER OF THE RECORD ** **REM CLOSED ** **IEH TEHDIA INFEGE ** ***TEH PORRY INFEGE ** ***TEH PORRY INFEGE ** ***TEHPORRY INFEGE ** **SIMITYPE = (/ STHIUNII(\$ACGES* **SIMITYPE = (/ STHIUNII(	ASSIGNS THE LINES AND BASIGNS THE LINES AND BASIGNS THE LINES AND BASIGNS THE LINES AND BASIGNS THE LINES AND RECORDS. IT CALLS PLACE HARCELS TO FILE 3 RECORD. RECORD		
**************************************		000 000 000 000 000 000 000 000 000 00	**3(TEM REC):) /) ;  ***********************************	*SIMITYPE = ( STMT UNIT(\$ACCES* *SIMITYPE = ( STMT UNIT(\$ACCES* *SIMITEM*REC)\$ /) \$  **SIMITEM*REC)\$ /) \$  **SIMITEM*REC)\$ // \$  **SIMITEM*RECORD  **MAS BEEN CLOSED **	*** TEMPORARY INTEGER **  ** TEMPORARY INTEGER **  *** TEMPORARY INTEG	**************************************	* BEING CLOSEON NUMBER OF THE RECORD * * BEING CLOSEON** * IIEM TEMPLATI INTEGER * * ITEM TEMPLATI INTEGER * * ITEM TEMPLATI INTEGER * * ITEM TEMPLATI INTEGER * * SITEM TEMPLATICATE THAT THIS RECORD * * MAS BEEN CLOSED **	** THE PROCEDURE CLOSE*REC  ** SSIGNS THE LINES AND  ** ALCK WIDTH FIELDS OF FILE 3  ** RECORDS.** IT CALLS PLACE.**  ** WHICH COMPLETES THE PART 1  ** PROCESSING OF THE FILE 3  ** RECORD.**  ** THE PART 2 STHT TYPE OF THE*  ** THE PART 2 STHT THE RECORD.**  ** THE PART 3 STHT THES RECORD.**  ** THE PART 4 STHT THIS RECORD.**  ** THE PART 5 STHT THIS RECORD.**  ** THE PART 5 STHT THIS RECORD.**  ** THE PART 5 STHT THIS RECORD.**	THE RESIDENCE OF THE PARTY OF T	The state of the s
HAVE A CONTROL BOX HUST HAVE AN OOD NUMBER			*53(TERM REC)\$) /) \$	*SIMI'IYE   SIMI'(SACCES**SIMI'(SACCES**SIMI'I	*ITEM TEMPIZ INTEGER * * * * * * * * * * * * * * * * * * *	*ITEM TEMPIA INFEGER ** *ITEM TEMPIA INFEGER ** *ITEM TEMPIA INFEGER ** *ITEM TEMPIA INFEGER ** *SIMITYPE = (///////////////////////////////////	**RECORO NUMBER OF THE RECORD ** **REING CLOSEO* **IEH TEMDIA INFEGER ** **TEM PEMDIA INFEGER ** **TEM PEMDIA INFEGER ** **TEM PEMDIA INFEGER ** **SAITE HYPE = (/ SIMIA UNII (**ACGES** **SAITE HYPE = (/ SIMIA UNII UNII UNII UNII UNII UNII UNII UN	ASSIGNS THE LINES AND  ASSIGNS THE LINES AND  BLOCK HIDTH FIELDS OF FILE 3  RECORDS. IT CALLS PLACE.  WHICH COMPLETES THE PART 1  PROCESSING OF THE FILE 3  RECORD  THE PART 2 STHT TYPE OF THE **  *** THE PART 2 STHT TYPE OF THE **  *** THE PART 2 STHT TYPE OF THE **  *** THE PART 2 STHT TYPE OF THE **  *** THE PART 2 STHT TYPE OF THE **  *** THE PART 2 STHT TYPE OF THE **  *** THE PART 2 STHT TYPE OF THE **  *** THE PART 2 STHT TYPE OF THE **  *** THE PART 2 STHT TYPE OF THE **  *** THE PART 2 STHT TYPE OF THE **  *** THE PART 2 STHT TYPE OF THE **  *** THE THE PART 3 STHT TYPE OF THE **  *** THE THE PART 3 STHT TYPE OF THE **  *** THE THE PART 3 STHT TYPE OF THE **  *** THE THE PART 3 STHT TYPE OF THE **  *** THE THE PART 3 STHT TYPE OF THE **  *** THE THE PART 3 STHT TYPE OF THE **  *** THE THE PART 3 STHT TYPE OF THE **  *** THE THE PART 4 STHT TYPE OF THE **  *** THE THE PART 4 STHT TYPE OF THE **  *** THE THE PART 5 STHT TYPE OF THE **  *** THE THE PART 5 STHT TYPE OF THE **  *** THE THE PART 5 STHT TYPE OF THE **  *** THE THE THE THE TYPE OF THE **  *** THE THE TYPE OF THE TYPE OF THE **  *** THE THE TYPE OF THE TYPE OF THE **  *** THE THE TYPE OF THE TYPE OF THE **  *** THE TYPE OF TYPE OF THE TYPE		. "INDICATE THAT THIS RECORD " HAS BEEN CLOSED ".
HUST HAVE A CONTROL BOX HUST HAVE AN OOD NUMBER			#S3 (TERM REC)E) /) 3	SOINT TYPE ALL STHICKS ** *SINT TYPE ALL STHICKS CES* *SINT TYPE ALL STHICKS CES* *SINT TYPE ALL STHICKS CES*	#ITEM TEMPIZ INTEGER # # #STMT TYPE = ( STMT (SACCES# #SJMT TYPE = ( STMT	*ITEM TEMPLI INFEGER ** ***TEM PADARY INFEGER ** ***TEMPORARY INFEGER ** ***TEMPORARY INFEGER ** **SIMT*TYPE =(/ STMT*UNIT(\$ACCES* **S3(TEM**REC)E) /) \$	**RECORO NUMBER OF THE RECORD ** **REMOTOR CLOSED* **IEM TENDIA INFEGR ** ***TEM PORARY INFEGR ** ***TEM POPARY INFEGR ** ***TEMPORARY INFEGR ** **TEMPORARY INFEGR ** ***TEMPORARY INF	ASSIGNS THE LINES AND  ASSIGNS THE LINES AND  BLOCK HIDTH FIELDS OF FILE 3  RECORDS, IT CALLS PLACE,  HHICH COMPLETES THE PART 1  PROCESSING OF THE FILE 3  RECORD  ITEM STHITYPE INTEGER 8  **RECORD NUMBER OF THE RECORD.**  RECORD  ITEM TEMPLE INTEGER 8  **RECORD.NUMBER OF THE RECORD.**  RECORD  ITEM TEMPLE INTEGER 8  **ITEM TEMPLE = 1 / STHITUMIT (\$ACGES**  **SITEM TEMPLE = 1 / STHITUMIT (\$ACGES**		Versions and service of the service
HUST HAVE A CONTROL BOX HUST HAVE AN OOD NUMBER				***TEMPORARY INTEGER ** *STHITTPE =(/ STHI"UNIT(\$ACCES*	*ITEM TEMPLE INTEGER \$  ****TEMPORARY INTEGER **  *STMI*TYPE = (/ SIMT*UNIT(\$ACGES*	*ITEM TEMPIT INFEGER **  ***IEM PORRAY INFEGER **  *ITEM TEMPIT INFEGER **  ****TEMPORRAY INFEGER **  ****TEMPORRAY INFEGER **  ***TYPE =(/ SIMI*UNIT(\$ACGES*	**RECORD NUMBER OF THE RECORD *  * BEING CLOSED**  * IEM INTEGER *  * TEM PORRY INTEGER *	ASSIGNS THE LINES AND BACKWHIDTH FIELDS OF FILE 3 RECORDS. IT CALLS DLACE. HHICH COMPLETES THE PART 1 PROCESSING OF THE FILE 3 RECORD.  "ITEM STHT TYPE INTEGER 8 ""FECORD NUMBER OF THE RECORD "" ITEM TEMPTE INTEGER 8 ""FECORD NUMBER OF THE RECORD "" ITEM TEMPTE INTEGER 8 ""TEM TEMPTE INTEGER 8		SS (TERM*REC)E) /) B
HAVE A CONTROL BOX HUST HAVE AN OOD NUMBER			αΞ w: w:	«¥ .:	* RECORDS** *ITEM TEKN*REC INTEGER \$ ***RECORD NUMBER OF THE RECORD * * BEING CLOSED**	* RECORD**  * ITCH FEC INIEGER \$	• 000000	SE REC VO OF FILE ACE, ILE 3		10 E
HAVE A CONTROL BOX HUST HAVE AN OOD NUMBER			TYPE OF THE RECO	TYPE OF	RECO	10 E	10 E	ASSIGNS THE LINES AND BLOCK HIDTH FIELDS OF FILE 3 RECORDS, IT CALLS PLACE, HHICH COMPLETES THE PART 1 PROCESSING OF THE FILE 3 RECORD. **		TIEM STATE TYPE INTEGER & *
HAVE A CONTROL BOX HUST HAVE AN OOD NUMBER			SER SECOF	SER S TYPE OF THE RECO	S OF	. O	. OF	THE PROCEDURE CLOSE REC SIGNS THE LINES AND COCK HIDTH FIELDS OF FILE CORDS, IT CALLS PLACE, ICH COMPLETES THE PART 1, OCESSING OF THE FILE 3		•
HAVE A CONTROL BOX HUST HAVE AN OOD NUMBER			SER S TYPE OF THE RECO	TYPE OF	S OF	0.5	0.0	THE PROCEDURE CLOSE REC SIGNS THE LINES AND OCK WIDTH FIELDS OF FILE CONDS. IT CALLS PLACE. ICH COMPLETES THE PART I OCESSING OF THE FILE 3		
HAVE A CONTROL BOX HUST HAVE AN OOD NUMBER			**************************************	SER SIYPE OF	RECO		0.5	THE PROCEDURE CLOSE'REC SIGNS THE LINES AND OCK'HIDTH FIELDS OF FILE CORDS, IT CALLS PLACE,		* MAICH COMPLETES THE PART I * PROCESSING OF THE FILE 3
HUST HAVE A CONTROL BOX HUST HAVE AN OOD NUMBER			FILE 3	FILE 3 SER S TYPE OF THE RECO	LE 3	LE 3	PART ILE 3	THE PROCEDURE CLOSE*REC SIGNS THE LINES AND OCK* HIDTH FIELDS OF FILE		
HUST HAVE A CONTROL BOX			PLACE.  HE PART 1.  FILE 3.  YPE OF THE RECORD.	FILE 3 FI	ACE. PART 1 LE 3 S OF THE	ACE. S. LE S. L. S. C. S	PART 1 ILE 3 ILE 3 S OF THE	THE PROCEDURE		FILE
HUST HAVE A CONTROL BOX			S AND S OF FILE S HE PART 1 FILE 3 FILE 3 FI	S AND SS OF FILE S OF FILE HE PART 1 FILE 3 FILE 5 FILE 3 FILE 5 FILE 5	ACE	ACE. ACE. PART 1. LE 3. S OF THE	ACE. ACE. LE 3 LLE			THE PROCEDURE
HUST HAVE A CONTROL BOX			S AND S OF FILE S PLACE: S PLACE: FILE 3 FILE 3 FIL	S AND SS OF FILE SS OF FILE SHE PART 1 FILE 3 FILE	PART 1 LE 3 LE 3 CF THE	SERECTO SERVICE STREET	SE'REC NO FILE ACE, PART 1 ILE 3			
HUST HAVE A CONTROL BOX			LLOSE REC S AND SS OF FILE S PLACE, FILE 3 FILE 3 FILE 3 FILE 3 FILE 3 FILE 3	CLOSE REC S AND S OF FILE THE PART 1 FILE 3	E REC D FILE D FILE PART 1 LE 3 LE 3 C F THE	SEREC SEREC ACE:	SE REC ND FILE ACE. ILE 3			

C S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF CLOSE\*REC

•	FERN'REC'S) TO C'ERN'REC'S) TO C'ERN'REC'S) TO C'ERN'REC'S)			
*IF TEMPIZEQ 0 \$ *+	**************************************		LINES(\$ACCESS3(TERM'REC)\$) +*  LINES(\$ACCESS3(TERM'REC)\$) +*  2 \$  ********************************	ARSOLS STUDY (TERM ARSORS STATES STAT
		ADO LINES FOR BOX EDGES AND DOUBLE SPACE OPTION	IFEITH ************************************	** PLACE (TERM*REC) & * * * * * * * * * * * * * * * * * *

```
*********
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      *---MESSAGE = 11H(COMPUTE NOS) & *
*OUT(MESSAGE, MESS'SM) & *
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       ************************
                                                                                                 *** THE PROCEDURE
COMPUTE PAGE NUMBERS ASSIGNS
REFEPENCE NUMBERS TO STUMPS
AND IF HEADING AND
TABLE OF CONTENTS ARE ON.
CALLS GENERATE "HEADERY ENTRY
TO ENTER A MODULE NAME INTO
THE TABLE OF CONTENTS. "
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 .IF HEADING AND TABLE OF CONTEN .
                                                                                                                                                                                                                                                                                     ***NUMBER OF PAGE ON WHICH
** MODULES START** **NUMBER OF P***ITEM CONTENTS*PAGES INTEGER $ ****PAGES SPANNED BY CONTENTS **** TABLE**
                                                                                                                                                                                                                                                                                                                                                                                             *ITEM CUM* PAGES INTEGER $
***CUMULATIVE PAGE COUNT**
*ITEM STUMP*COUNT INTEGER P 0 $*
                                                                                                                                                                                                                                                                                                                                                                                                                                                                ***CUMULATIVE STUMP COUNT **
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     **************************
                                                                                                                                                                                                                                                                           *************************
                 *PROC COMPUTE*PAGE*NUMBERS $ *
*****************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       *CUR* GROUP = NEXT(30$) $ *
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        .......................
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             *****************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           *CUM*PAGES = PAGE*NO $ *
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        * IF DEBUGI $ .
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          ************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      *************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        ....
```

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER DESIGN DIAGRAM OF COMPUTE" PAGE" NUMBERS

A S S S S S S S S S S S S S S S S S S S	** NUMBER STUMPS ACCORDING TO ORDER OF APPEARANCE *** RATHER THAN BY PAGE NUMBER	STUMP COUNT = STUMP COUNT + 1+  * PAGE REF(SCUR GROUPS) =   * STUMP COUNT \$	1F DEBUG25 8 39 4 39	
THEADING & CONTRACT OF CUMPAGES + 1 & COMPAGES = CUMPAGES + 1 & PAGE REF(SCUR GROUPS) & PAGE REF(SCUR GROUPS) = 1 & ASS PAGE NO & STATE PROCURS OF CONTRACT AND THE TABLE OF C	+-ORIF PROTURHE(SCUR'GROUPS) EQ + + SPACES(VAXCOL) & +	PROPERTY OF THE PROPERTY OF TH	11 DE	*CUR*GROUP = NEXT(SCUR*GROUPs)*
E E E E				*CUR* GROUP
DO WHILE CCUR'GROUP GR 0 1				

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER DESIGN DIAGRAM OF COMPUTE"PAGE"NUMBERS

** ** ** ** ** ** ** ** ** ** ** ** **			
	*****		
		(A)	PARK SK MAND MANUAL PROPERTY OF
			1000
		A VENEZA CARA CARA CARA CARA CARA CARA CARA CA	And the state of t

\*\* THE PROCEOURE CONNECT\*BOXES

OETERHINES HOW TO CONNECT A

CODE BLOCK TO ITS FATHER (IF

IT HAS ONE). IF THE CODE

BLOCK IS A STUHP THEN

CONNECT\*BOXES DRAWS THE STUMP

REFERENCE DISPLAY BOX FOR THE

TITEN DISPLAY TOP INTEGER \$

"ITEM FATHER INTEGER \$

"RECORD NUMBER OF CODE

"BLCCK'S FATHER."

"ITEM FATHER" BOTTOM LINE

"ITEM FATHER" BOTTOM LINE

"ITEM FATHER" TOP LINE

"ITEM LINGTH INTEGER \$

"ITEM PAGE INTEGER \$

TITEM XI INTEGER \$

"" X COORD OF CONNECTING LINE'S"

START"

LINE XZ INTEGER \$

"" X COORD OF CONNECTING

LINE YZ INTEGER \$

"IFM YI INTEGER \$

"IFM YI INTEGER \$

"IFM YI INTEGER \$

"IFM YZ INTEGER \$

"ITM YZ INTEGER \$

"

\*IFEITH \*-+-\*PAGE\*RE\*(\$CUR\*GROUPS) LS 10 \$ \*---\*LNGTH = 1 S \* . 1. 5 \*IF HEADING \$ \*\*\*\*\*PAGE" REF (\$CUR"GROUPS) = PAGE +\* \* \* \*REMONO(LIME'NO, PAGE'LNGTH == PAGE, LIME) \$ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of Connect'Boxes \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* +--ORIF 1 S -----\*T\*MESS = IFORMAT(FATHER) \$ \*\* \* \*T\*HESS # 4HI TO 1 \$ \*MESSAGE + T\*MESS)\* \*OUT(MESSAGE, MESS\*SA) \$ \*MESSAGE = CATIMESSAGE, T.MESS) \* \*\*\*\*\*\*\*\*\*\*\*\*\* \*T MESS = IFORMAT (CUR'REC) \$ \*\*\*\*\* MESSAGE = BHICONNECT ) \$ \*\*\*\*\*\*\*\* \*RETURN \$ . \*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\* . IF DEBUGS \$ . \*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\* ....... \*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF CONNECT BOXES

	ORIF PASE REFISCUR GROUPS) LS	4 7 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -	-	THE RESIDENCE OF THE PARTY OF T	在这条中的大学中的大学的一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个		E.REFISCUR GROUPS) LS .		电路线路路线电路路路线电路路路线电路路路路路路路路路路路路路路路路路			+OKIT I WKK (BH(HHOOPS) S +		*DISPLAY*TOP = TOP*LINE -	* V*SPACE - 3 S	*STARS(HARGIN, OISPLAY*MIDTH.*	* DISPLAY*TOP) \$	法操作的 医格特氏征 医克格特氏征 医克格特氏征 医克格特氏征 医克格特氏征 医克格氏征 计数据		TOTAL STATE					1		47 (#H1C170-XV   GOULD   N1COVER 0 / 11 / 12 / 12 / 12 / 12 / 12 / 12 /	# 1	# CONTROL   CONT	*TEMPC = IFORMAT (PAGE REF (SCUS"*	\$ 639UP\$3.	*BYTE(\$MARGIN + 2.	• LNGTH\$) (OUT • LINE (\$ACCESS • ) UT ( •	*DISPLAY*TOP + 1)\$)) = BYTE(\$6,*	* LNGTH\$)(TEMPC) \$	•		!		*(AY*TOP + 1)\$)) = 4H(FROM) \$ *	TOTAL STORY		A COLUMN TIONAL OF THE PARTY OF	A WW
		•	•	•	•	•	•	•	•	•			•	•	•	•		•		•	•	•	•	•	•	 	* Cardolla La		• •	•	•	•	• 1	•	•	 •	•	•	•	•	OFFICE OR CERSE	the transfer out the	•	

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of connect\*boxes

* LNGTH\$) (OUT*LINE (\$ACCESS*OUT(* *OLSPLAY*TOP * 1)\$) = BYTE(\$6,* * LNGTH\$) (TEMPC) \$	***************************************	*STARS(MARGIN, DISPLAY*MIDTH.*	10 SPLAY 100 + 2) 8	HAKGIN + 1 S	11   11   12   12   12   13   14   15   15   15   15   15   15   15				of ATHER' BOTTOM =	* STOP*LINE (\$ACCESS3(FATME2)\$) *	\$ 4010	PATE STATE OF THE	*	* SIMI*UNII (BACCESSG (FAIHER) #	· A settle for a representative of the section of t		 STATETH STREETS OF STR		***************************************	•			电电话电话电话电话电话电话电话电话电话电话电话电话电话电话电话电话电话电话电	***************************************	Securification Section State of the FO CONTROL S C Section 1		*** *** ** ** *** *** *** *** *** ***	evi = FATHER-BOTTO4 S	を受けています。
•••						•																							

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF CONNECT"BOXES

PAGE ..

---ORIF 1 S ----KZ = LEFT-COL + 1 S + OY2 = TOP-LINE S + OY2 = TOP-LIN

Restriction of the control of the co

NEIGHT DINGER OF CONTRACT SERVICE OF SECUM SECUMENDS

LINE EMBEDS THE LINE OF INPUT
LINE EMBEDS THE LINE OF INPUT
EXT PACULOED BY THE
PROCEDURE EXTRACT: TEXT IN THE
SIDES OF THE OF THE BOX IN
HIGH IT MILL APPEAR.
HIGH IT MILL APPEAR.
CONSTRUCT: LINE ALSO IS
RESPONSIALE FOR OUTPUTING
BLANK TEXT LINES IF THE
DOUBLE 'SPACE OPTIONS IS IN
EFFECT: DOUBLE SPACING IS
HANDLED BY ITERATING THROUGH
THE ROUTINE INC ITHES WHERE
INC IS SET ACCORDING TO THE
VALUE OF SKIP, THE BLANK LINE
FLAG. \*PROC CONSTRUCT\*LINE S \*

\*ITEM PAGE INTEGER \$
\*\*\*OUNHY VARIABLE USED FOR CALL\*
\* TO REMQUO\*\* FITTH REAL START INTEGER \$

\*\*STORES THE COLUMN IN MAICH \*\*

\*\* TEXT STARTS\*\* \* \*\*\* THEM INC INTEGER \$

\*\*\*\* THE UPPPER BOUND FOR THE OUTPUT LOOP\*\*

\*\*\*\* TIEM LINE INTEGER \$

\*\*\*\* LINE NO'S DISPLACEMENT ON CURRENT PAGE\*\*

\*\*\*\* TIEM LNGTH INTEGER \$

\*\*\*\*\* LENGTHS\*\*\*

\*\*\*\* LENGTHS\*\*\*

. IF DE BUG9 8 .. \*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*

\*LNGTW = LENGTH(NEW TEXT) & \*

C S DRAMER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANYER DESIGN DIAGRAN OF CONSTRUCT'LINE

ST SUNIO S dIMS+	S S S S S S S S S S S S S S S S S S S	TO INC 8 TINE, NO = TINE, NO + I 8 +	TF HEADING S+  "IF HEADING S+  "HEADING "."  "HEADING "."	**************************************	**************************************	**************************************	TEMPC = OUT-LINE(\$ACCESS*OUT(L*  * INE*NO'S) S  **********************************	erreces establisher EQ COMMENT'S \$ ***********************************	+-ORIF STHT TYPE EQ SEQ OR + **********************************
S disco.	T JINO	FOR I = 0. I. INC s							

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of Construct" Line

TEITH TIPE CQ CONTROL : STEETH TO STATE ST
--

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design Diagram of Construct"Line

						-	
					-		
					-		-
					- 3		-
					-	_	-
					-		
					- 1		
					-	5	
					- 2	-	-
						=	
						I	
40	•	-				9	
		-					
46	_					œ	
	ž						
	4					-	
	_	•					
	Ħ					S	
	2					0	
3	×	٠				0	
ж	٠						
•						-	
						I	
ж	0					9	
	SU FROM 46	**************			************************	*RIGHT POS = RIGHT POS + 1 \$ *	*******************
						œ	
		-		-	-	-	-
×							

C. S. DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DÍAGRAMMEN DESIGN DIAGRAM OF CONSTRUCT°LINE

DESTRUCTION OF THE CONSTRUCT LINE
* 51 FROM 48 *
• • • • • • • • • • • • • • • • • • • •
*IF LINE NO EQ TOP-LINE + 1 \$ **** BYTE(SLEFT*COL, 18)(TENPO) = *
*DASHES(LEFT*COL * 1. BOX*TAIL*
***************************************
***************************************
*REAL START = LEFT*COL + * * BOX*TAIL + 1 3 *
*BYTE(SLEFT*COL + BOX*TAIL.* * 15)(TEMPC) = 1H(*) s *
***************************************

C S ORAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design Diagram of Constructione

\* 52 FROM 48 \*

THE PROCEDURE CONTINUE BOX APPENDS COMPATIBLE STATEMENT UNITS TO THE CURRENT FILE 3 RECORD. A COMPATIBLE STATEMENT CAN APPEND TO THE CAME CODE BLOCK NITH THE STATEMENT UNITS WHICH HAVE ALREADY BEEN PLACE THERE.  THE CONTINUE BLOCK.  THE CONTINUE BLOCK.  THE SAGE = \$H(CONT) \$  THE SAGE = \$H(CONT) \$	*THE PROCEDURE CONTINUE*BOX
THE PROCEDURE CONTINUE BOX  APPENDS COMPATIBLE STATEMENT UNITS TO THE CUPRENT FILE 3  RECORD. A COMPATIBLE STATEMENT UNIT IS ONE WHICH CAN APPEND THE SAME CODE BLOCK WITH THE STATEMENT UNITS WHICH HAVE ALREADY BEEN PLACED THERE  INTILATED BOX  "CONTINUE A PREVIOUSLY INTILATED BOX "PESSAGE = 5H(CONT) & "THESS = IFORMATICONT REC) & "STATEMENT CONTINUED BLOCK."  "CONTINUE A PREVIOUSLY INTILATED BOX "PESSAGE = GAT(HESSAGE, THESS  "CONTINUE A PREVIOUSLY INTILATED BOX "THESS = IFORMATICONT REC) & "STATEMENT CONTINUED BLOCK." "STATEMENT CONTINUED."	THE PROCEDURE CONTINUE.BOX
**THE PROCEDURE CONTINUE BOX  **APPENDS COMPATIBLE STATCHENT UNITY TO THE CURRENT FILE 3  **RECORD. A COMPATIBLE STATEMENT UNIT IS ONE WHICH CAN APPEAR IN THE SAME CODE BLOCK WITH THE STATEMENT UNITY MHICH HAVE ALREADY BEEN PLACEO THERE.  **THE CONT.REC INTESER 8  **THESS ENCONT.**  **ITEM CONT.**  **	THE PROCEDURE CONTINUE.BOX
WAPPENDS COMPATIBLE STATEMENT UNITS TO THE CUPARANT FILE 3 RECORD. A COMPATIBLE STATEMENT UNIT IS ONE WHICH GLOK WITH THE SAME CODE BLOK WITH THE STATEMENT UNITS WHICH HAVE ALREADY BEEN PLACE DITHER.  "CONTINUE A PREVIOUSLY INITIATED BOX ""CONTINUE A PREVIOUSLY INITIATED BOX ""TOMESS = IFORMATICONT REC) 8 ""S" """ """ """" """" """"""""""""""	
PLACEON A COMPATIBLE  STAFFENT UNIT IS ONE WHICH  CAN APPEAR IN THE SAME CODE  BLOCK MITH THE STATEMENT  UNITS WHICH HAVE ALREADY BEEN  PLACED THERE  TEM CONT'REC INTESE S.  TIEM CONT'REC INTESE S.  TYPESSAGE SHICONT'REC) S.  THEST SAGE SHICONT'REC) S.  THE STATEMENT SHICONT'REC'S S.  THE STATEMENT SHICONT'REC'S SHICONT	* APPENDS COMPATIBLE STATEMENT
**RECRRD. A COMPATIBLE  STATEMENT UNIT IS ONE WHICH GAN APPEAR IN THE STATEMENT  BLOCK MITH THE STATEMENT  UNITS WHICH HAVE ALREADY BEEN  PLACED THERE.  **REC NO. OF CONTINUED BLOCK.**  **REC NO. OF CONTINUED BLOCK.**  INITIATED BOX **  **THESS = FHCONT REC) **  **LINES(BACCESSICONT REC) **  **L	. UNITS TO THE CUPRENT FILE 3
CAN APPEAR IN THE SAME CODE  BLOCK WITH THE STATEHENT  UNITS WHICH HAVE ALREADY BEEN  PLACED THERE  TEC NO. OF CONTINUED BLOCK  "CONTINUE A PREVIOUSLY  INITIATED BOX "  "T'MESS = IFORMATICONT'REC) S  "T'MESS = I	* RECORD. A COMPATIBLE
**CAN APPEAR IN THE SAME CODE  **BLOCK WITH THE STATEHENT  **UNITS MICH HAVE ALREADY BEEN  **PLACED THERE.**  **ITEH CONT'REC INTEGER \$  ***********************************	IENT UNIT IS ONE
"CONTINUE A PREVIOUSLY INITIATED BOX "CONTINUE B PREVIOUSLY INITIATED BOX "TO BE SAGE = 5H(CONT ) &  "CONTINUE A PREVIOUSLY INITIATED BOX "TO BE SAGE = FH(CONT ) &  "TO BUGGO & "TO BE SAGE = FH(CONT ) &  "TO BE	PPEAR IN THE
"CONTINUE APENTANE ALREADY BEEN  "EEC NO. OF CONTINUE BLOCK."  "ECCONTINUE A PREVIOUSLY  INITIATED BOX "  "THESS = IFORMATICONT'REC) &  "THESS = IFORMATICON	WITH THE STA
"ELACED THERE."  "INTITATES BOX"  "CONTINUE A PREVIOUSLY  INTITATES BOX"  "THESS = IFORMATICONT"REC) S  "THESS = IFORMATICONT"REC) S  "THESSAGE = GAT(HESSAGE, THESS) S  "THESSAGE, HESSAGE, THESSAGE, THESSAG	WHICH HAVE ALREADY
"" REC NO. OF CONTINUED BLOCK."  ""CONTINUE A PREVIOUSLY INITIATED BOX "" T'MESS = IFORMATICONT'REC) & "T'MESS = IFORMATICONT'REC) & "METHES = IFORMATICS = I	D THERE
TIEH CONT.REC INTESER S  ""REC NO. OF CONTINUED BLOCK."  INITIATED BOX ""  INITIATED BOX ""  IN INITIATED BOX ""  """  """  """  """  """  """  """	
"" REC NOT REC INTESER & "" REC NO. OF CONTINUED BLOCK."  "" CONTINUE A PREVIOUSLY  INITIATED BOX " "THESSAGE = 5HCONT ) & "THESSAGE = 5HCONT ) & "THESSAGE = 6THESSAGE, THESSAGE, THESSAG	
""CONTINUE A PREVIOUSLY  ""CONTINUE A PREVIOUSLY  INITIATED BOX ""THESS = IFORMATICONT"REC) \$ "FY-END (\$ACCESS!(CONT"REC) \$ "FY-END (\$ACCESS!(CONT"REC) \$ "LAST" 4, \$ "LINES!ACCESS!(CONT"REC) \$ "LINES!ACCESS!(CONT"REC) \$ "LINES!ACCESS!(CONT"REC) \$ "LINES!ACCESS!(CONT"REC) \$ """ LINES!ACCESS!(CONT"REC) \$ """ LINES!ACCESS!(FIRST" """ LINES!ACCESS!(CONT"REC) \$ """ LIN	*************
""CONTINUE A PREVIOUSLY INITIATED BOX ""  ""CONTINUE A PREVIOUSLY INITIATED BOX ""  "T" "F" "T" ""  "T" "" "" "" "" "" "" "" "" "" "" "" ""	
"CONTINUE A PREVIOUSLY INITIATED BOX "  "INITIATED BOX "  "INITIATED BOX "  "THESS = FHCONT   S    "THESS = IFORMATICONT*REC) S    "THESS = IFORMATICESS S ICONT*REC) S    "THESS S IFORMATICES S ICONT*REC) S    "THEST S IFORMATICES S ICONT*REC) S    "THESS S IFORMATICES S ICONT*RECS S	CONTINUED
""CONTINUE A PREVIOUSLY "INITIATED BOX"  INITIATED BOX"  INITI	
INITIATED BOX "  INITIATED BOX "  INITIATED BOX "  IF DEBUG10 \$ " "T'MESS = FHCONT 'S STORE TO STORE T	************************
INITIATED BOX "  INITIA	
"."CONTINUE A PREVIOUSLY "INITIATED BOX".  "INITIATED BOX".  "INITIATED BOX".  "IN HESSAGE = 5H(CONT) &  "I'MESSAGE = 6AT(MESSAGE, T*MESSAGE, T	
INITIATED BOX **  INITIATED BOX **  **  **  **  **  **  **  **  **  **	
INITIATED BOX "  "THESSAGE = GH(CONT ) \$  "THESS = IFORMATICONT P.E.) \$  "THESSAGE = GAI(HESSAGE, THESSAGE, THESS) \$  "OUTHESSAGE, HESSAGE, THESSAGE, THESSA	
"IF DE BUG10 \$ """ MESSAGE = \$H(CONT) \$ """ PMESSAGE = \$H(CONT) \$ """ PMESSAGE = \$TOTHESSAGE, """ PMESSAGE = CAT(MESSAGE, """ PMESSAGE = CAT(MESSAGE, """ PMESSAGE = CAT(MESSAGE, """ PMESSAGE = CAT(MESSAGE, """ PMESSAGE, "" PMESSAGE, """ PME	
TIF DEBUGIO \$ * **T*******************************	
"IF DE BUG10 \$ """ """ """ """ """ """ """ """ """	
"TF DE BUG10 & """ HESSAGE = GAT(HAZIGONT) & """ HESSAGE + """	***********
**************************************	DEBUGIO &
**************************************	
### ##################################	
"ESSAGE, MESSAGE, PESSAGE, PES	
E. HESS-SH)	
	DOTTHE STATE + HE SS SW) S
** FATOR SACCESSICONT** REC)\$) = ** LANT** & ** LANT**	
*Ft*ENDISACCESS3(CONT*REC)\$) = * *LAST*4 *LINES(LACCESS3(CONT*REC)\$) = * *LINES(LACCESS3(CONT*REC)\$) + * *LINES(LACCESS3(CONT*REC)\$) + * *LINES(LACCESS3(CONT*REC)\$) + * *LINES(LACCESS3(CONT*REC)\$ + * *S	************
LAST 48  *LINES(BACCESS3(CONT*REC)\$) + *  *LINES(BACCESS3(CONT*REC)\$) + *  *LINES(BACCESS3(CONT*REC)\$) + *  *LINES(BACCESS3(CONT*REC)\$) + *  *BLOCK*WIDTH(\$ACCESS3(CONT*REC)\$ + *  *GCESS4(FIRST*4)\$)) \$  *RETURN \$  *RETURN \$	A L SELLEGIC CONTOCULATION - 1 TH
*LINES(BACCESS3(CONT*REC)\$) = * *LINES(BACCESS3(CONT*REC)\$) + * *LINESSOUT(\$ACCESS4(FIRST*4)\$)* *BLOCK*WIDTH(\$ACCESS3(CONT*REC)* *\$) = MAX(BLOCK*WIDTH(\$ACCESS3(***CONT*REC)** *CONT*REC)\$; *MAX*LINE*LNGTH(\$A***CONT*REC)\$; ** **RTURN \$ **RTURN \$	
LINESACCESSICONTRECTS:  * LINESCOUT(\$ACCESSICONTRECTS) + *  * LINESCOUT(\$ACCESSICONTRECTS) + *  * LINESCOUT(\$ACCESSICONTRECTS) + *  * THAN (\$ACCESSICONTRECTS)	THE PARTY OF THE P
LINES GOUT (\$ACCESS& (FIRST 4) \$) **  * LINES **  * BLOCK ** NID TH (\$ACCESS (CONT * REC) **  * S	TANGET STATE OF THE STATE OF TH
* SELOCK*WIDTH(SACCESSACONT*REC)*  **BLOCK*WIDTH(SACCESSACONT*REC)*  **S) = MAX(BLOCK*MIDTH(SACCESSAC***CCESS*(**TCESS*********************************	# 1 TAIL STANFOLD SOUTH AND THE PARTY OF THE
*BLOCK*WIDTH(\$ACCESS3(CONT*REC)*  *\$) = MAX(BLOCK*WIDTH(\$ACCESS3(* *CONT*REC)\$), MAX*LINE*LNGTH(\$A* *CCESS4(FIRST*4,)\$)) \$  *RETURN \$  *RETURN \$	LINES UNITRACLESSALFIRST 4787
*BLOCK*WIDTH SACCESSICON'*REC)*  **) = Max(BLOCK*WIDTH(SACCESSIC*  **ON**REC)*), MAX*LINE*LNGTH(SA*  **CCESSIC****  **RETURN 3	
TATION S TRUE S TR	*BLOCK ** ID TH (SACCESS3 (CONT * REC) *
*CONT KECISI* MAX'LINE LNOTH(SA**CCESS4(FIRST*4)\$)) \$ **MFIFG3 = TRUE \$ **RETURN \$	ST = MAX (BLOCK MIDIM SACCESSA)
*CCESS4(FIRS*4)3) \$ ** *WFITE3 = TRUE \$ ** *RETURN \$	CONT RECISO, MAX LINE LNGTH (SA.
*MFITE3 = TRUE \$ **	+CCESS4(FIRST*4)\$)) \$
220	** FILE 3 = TRUE \$
	*KETUKN \$

TITEM BYTES INTEGER S

"" NO. OF BYTES PROCESSED."

"ITEM CURPOS INTEGER S

"ITEM CURPOS INTEGER S

"ITEM CURPOS INTEGER S

"" THE CURRENT FILE I RECORD."

"ITEM LAST'BYTE INTEGER S

"" THE CURRENT FILE I RECORD."

"ITEM LAST'BYTE INTEGER S

"" REC OF LAST BREAK."

"ITEM LAST'BYTE INTEGER S

"" REC OF LAST BREAK."

"ITEM LAST'BYTE INTEGER S

"" PART 2 STATEMENT TYPE OF

"" PART 3 STATEMENT

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANMER Design Diagram of Create-File4-Recs

THE PARTY OF THE P	
***************************************	
elf DEBUGLIS	
***7** **** MESSAGE = 9HICREATE 4 ) \$	
* TOMESS = IFORMATILAST" + 1) S*	
* ** ** ** ** ** ** ** ** ** ** ** ** *	
· · · · · · · · · · · · · · · · · · ·	
. SESSES HESSASE. NESS SH) S	
***************************************	
· · · · · · · · · · · · · · · · · · ·	
***************************************	
*FILE3*INCLUSION = TRUE \$ *	
*** ASSUME IT WILL BE A FILES *	
• ENTRY ••	
*STMT TYPE = BOX MAP(STMT TOKEN*	
************************	
明 二十二十二十二十二十二十二十二十二十二十二十二十二十二十二十二十二十二十二十	

STATITIVE IS A PRINTING STATE UNIT. IF ITS LENGTH ... EXCEEDS ST.MAX, THEN BREAK IT UP. ...

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF CREATE-FILE4-REGS

LINES  A K IT  TEMP BYTE = LAST + 1 S  TEMP BYTE = LAST BYTE + 5T MAX  TEMP BYTE GR MACOL S  THE ITHTEMP BYTE GR MACOL S  THE ITH	•••	• SZ!-1	
** INES OUTGAGGESSA(FIRST**)*) **  ** A ** LINE*LECKNESA(FIRST**) **  ** A ** LINE*LECKNESA(FIRST**) **  ** A ** STHT UNIT HUST SPAN LINES  ** SPACE**  ** IF NOT POSSIBLE, BREAK IT  A ** STHTACKNESS**  ** STHTA			
** ** ** ** ** ** ** ** ** ** ** ** **			
##X*LIME*LNGTH(\$ACCESS4(FIRST************************************	• •	+LINES OUT (SACCESS4 (FIRST 4) \$) +	
*418) = STHT LUGHT BE STATE LINES IR TO BREAK IT AT A SPACE IR NOT POSSIBLE. BREAK IT AT ST. HAX IF NOT POSS	•	* THE TRUE TREATMENT OF THE STOCK OF THE STO	
TRY TO BREAK IT AT A SPACE.  IF NOT POSSIBLE, BREAK IT AT STHAK  HAX LINE LNGTH (BACCESS4 (FIRST**  1) 18	•	S THOUSE THE COCTA	
TRY TO BREAK IT AT A  SPACE.  IF NOT POSSIBLE, BREAK IT AT ST HAX  HAX LINE LNGTH (BACCESS4 (FIRST**  1) 1) 10 10 10 10 10 10 10 10 10 10 10 10 10			
STATE UNIT HUST SPAN LINES  SPACE.  IF NOT POSSIBLE, BREAK IT  AT ST MAX  AT ST MAX  AT ST MAX  AT ST MAX  SOUNCE FRASE  CONNETE (NOT POSSIBLE)  CONNETE (NOT POSSIBLE)  CONNETE (NOT POSSIBLE)  SOUNCE FRASE  SOUNCE FRAS	ORIF 1 &		
SS4 (FIRST **  SS4 (FIRST **  **  **  **  **  **  **  **  **  **	***************************************	STHT UNIT HUST SPAN LINES	
SS4 (FIRST: SS4 (F		* TRY TO BREAK IT AT A	
SS4(FFRST:  SS4(FFRST:  TEMP'BYTE = LAST'4 + 1 S  TEMP'BYTE = LAST'BYTE + ST'MAX*  SUR'POS = 6YTES + ST'MAX S  TOUR'POS = 6YTES + ST		IF NOT POSSIBLE, BREAK IT	
SS4(FIRST:  LAST'4 + 1 S  TEMP'BYTE = LAST'8 + 1 S  CUR'POS = 6YTES + ST'NAX*  S  TONE POS = 6YTES + ST'NAX S  TONE FOR E + ST'NAX S  TONE POS = 6YTES + ST'NAX S		AT ST. MAX	
SS4(FFRST:  SS4(FFRST:  TEMP'BYTE = LAST'4 + 1 *  CUR'POS = 67TES + ST'HAX *  TEMP'BYTE			
TEMP'BYTE = LAST'4 + 1 S  TEMP'BYTE = LAST'BYTE + ST'HAX*  S  CUR'POS = 6YTES + ST'HAX S  TOUR POS = 6Y			
TEMP'BYTE = LAST'4 + 1 S  TEMP'BYTE = LAST'8 + 5 S  CUR.POS = 6YTES + 5T*MAX S  TO DOES THIS INCREMENT PUT US  IN NEXT FEREC **  TEETH ***-*********************************		# 19 = 0 \$ et 19	
TEMPORTE = LAST BYTE + ST MAX*  CURPDS = 6YTES + ST MAX & *  CURPDS = 6YTES + ST MAX & *  TEMPORTE = LAST BYTE + ST MAX*  IN NEXT F2REC *  IFEITH **			
TEMPORTE = LAST'4 + 1 S  TEMPORTE = LAST'8 YE + ST'MAX*  CURPOS = 6YTES + ST'MAX S  TO DOES THIS INCREMENT PUT US  IN NEXT FERC **  TEEITH **TEMPORTE GR MAXCOL S  ***********************************			
TEMPOSTE = LAST'6 + 1 \$  TEMPOSTE = LAST'BYTE + ST'MAX*  *CUR.PDS = 6YTES + ST'MAX \$  IN NEXT FERC **  *IFEITH *****  **IFEITH ****		******************	
S THIS INCREMENT PUT US  XT F2REC  TEMP BYTE GR HAKCOL 8  TEMP BYTE GR HAKCOL 8  TEMP BYTE GR HAKCOL 8		* *TEMP*BYTE = LAST*BYTE +	
S THIS INCREMENT PUT US XT FEREC		•	
STHIS INGREMENT PUT US XT F2REC		*CUR*DOS = GYTES + ST*MAX % *	
STHIS INCREMENT PUT US TERFORE GR MAXCOL 8 TEMPONTE GR MAXCOL 8	•		
XT F2REC **  ** F2			
TEMPOTE GRANCOLS		DOES THIS INCREMENT PUT US	
TEMPONTE GR HAXCOL S		电电子 化二甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基	
ORIF 1			.!
			•
			*TEMP*REC = LAST*REC + 1 8
.!		• •	**********************
!			
•			N. STARTER

FILE. RECS  SCAN BACK IN THE LINE  AN ACCEPTABLE GREAKPOINT  COD HHILE (CURPOS - BYTES G  LOW-LINE  CESSZ (TEMP'REC)3. TEMP'BY  LOW-LINE  CESSZ (TEMP'BY  LOW-LINE  CESSZ (TEMP'BY  LOW-LINE  TEMP'BY  SPECIAL OF THE LINE  TEMP'BY  TEMP'BY	TENDERS CECES CONTRACTOR CONTRACT	THE POST OF THE PO	65 .	NAME OF THE PARTY	OS = BYTES + ST*MAX \$ EYTE = LAST*BYTE + ST*MA	
M OF CREATE			1) NQ SPACES(1)]	BREAKPOINT BREAKPOINT		F2. BYTE(5ACESS4(LAST4)5) = 0 F2. PRE(5ACESS4(LAST4)5) = 0 F2. PRE(5ACESS4(LAST4)5) = 0 F2. PRE(5ACESS4(LAST4)5) = 0
	RAM OF CREATE-FILE4-REGS	••••••				

N DIAGRANGER	*** TRUE &	•••	JPDATE HAX LINE LNGTH	•	#4)&) = MAX(MAX*LINE*LNGTH(SACC*	*ESS4(FIRST*4)\$) . CUR*POS	PWRITE = TRUE				•	ANY MORE BREAKS REQUIRED					•	. 19		**********************		*HESSASE = CAT(MESSAGE, T.MESS) .	* * * * * * * * * * * * * * * * * * * *	***********************	
IAL STRUCTURED DESIGN FILE4.RECS		• •	••	•	•	•		•	•	•		•	•	•	•	•	•		***********	*IF DEBUG11 \$ .	 •	****			
C.S. DRAFER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMAER DESIGN DIAGRAM OF CREATE FILE4 RECS																									

• • • • • • • •

C S DRAFER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design Diagram of Create-File4-Recs

	Q 1 8 ****** ***************************	TEMPORE = TEMPORE - 1 S F FTEMPORE = MAXCO. S F FEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	**************************************
59 F40H 57 ·	TEITH *TEMPONTE EQ 1 S *		- OSIF I

KRANE SIGNAL GEORGE TELEFORMATION OF HERRY TREES WAS THE SERVER

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design Diagram of Create-Files-Recs

. 63 FROM 57 \*

\*TEMP\*BYTE = TEMP\*BYTE - MAXCOL\* \*

\*TEMP\*REC = TEMP\*REC + 1 \$

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANMER Design diagram of create-file4. Recs

FEITH "CURPOS + ST"MAX GQ SIMTLNGTH "  S		
94 YE E E E E E E E E	I FEITH G-G-CUR-POS + S	* PASKI THE SECTION OF
NEEDED AT LEAST 1 HORE NEEDED	\$	:
NEEDED AT LEAST 1 MORE NEEDED NE		• •
**************************************		
NEEDED "  NEEDED		
NEDED "  NEST 1 HORE  NEST 2 HORE  NEST 3 HORE  NEST 3 HORE  NEST 3 HORE  LAST REC = TEMP REC \$  LAST REC = TEMP REC \$  SHYTE		*LINES* DUT( \$ACCESS4 (FIRST* 4) \$)
NEEDED NEEDED NEEDED NEEDED NEEDER		. = .AST . + FIRST . + 1 8
NEEDED "  NEEDED		WAY LINE 'LNGTH (SACCESSA (FIRST
" CURPOS) & CURPOS) & CURPOS) & CURPOS) & CURPOS) & CURPOS) & CONE = TRJE & CONE = TRJE & CONE = TEMP'REC & CONE = TEMP'	•	TOWN TOWN TOWN TOWN TOWN TOWN TOWN TOWN
**************************************		* CU3*POS) \$
NEEDED "  NEEDED	• •	#DONE H TRUE &
NEEDED "  NEEDED	•	
NEEDED "  NEEDED	************	
**************************************		AT LEAST 1 MORE BREAK
**************************************	•	
** ** ** ** ** ** ** ** ** ** ** ** **	•	
* * * * * * * * * * * * * * * * * * *		*LAST*BYTE = TEMP*BYTE \$ *
		eLASI°REC = TEMP°REC \$ * * * * * * * * * * * * * * * * * *

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design Diagram of Create" H-Ptr-Rec

PROC CREATE H.PTR. REC(FATHER = +		EDURE 19-DEC SETS UP	IZ	ILE 3 TREE".	 ATHER INTEGER S ATHER OF THE NEM	SHER OF T		**************************************	* HESSAGE = CATCHESSAGE, T. HESS)	*OUTINESSAGE, MESS'SN) &		CESSS(FATHER) = SON		* = (\$(NOS)ES	5 0 H (\$(705)25	
* PROC CREATE	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	. THE PROCE	* POINTERS WHICH HAVE T	. INTO THE FIL	 TIEM FATHER INTEGES THE FATHER OF TH	E - 3	***************************************			•	SCN = F3.AVAIL	*H*PTR (SACCESSS(FATHER)S)	** S ** TRUE \$	*BACK*H(\$ACCESS3(SON)\$)	*BACK VISACCESS3 (SON)	*V*PTR (\$ACCESS3(SON)\$)

of Side Doresea on coffice size

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of Create-V-Ptr-Rec

● 等的發揮者者亦一正要以起去 目状神病不行之故或 谁好什么也仍然们故事	
:	
THE PROCEDURE CREATE	
REC IS THE	
PTR'REC EXCEP	
. IT CREATES A V'SON	
A STATE OF STREET STREET STREET STREET	
INTEGER S	
ORD NUMBER OF T	
TEM SON INTEGER \$	SAN CONTRACTOR
	2.00
err DE BUG12 &	**********
** MESSAGE =	E V ) \$
	3. AVAIL) \$
************ ** **********************	AGE. T. MESS)
99	
# EXC - CO IE - 019 4 CO IE - 100 - 1	S. M. S.
*****************************	
*SON = F3. AVAIL \$	
.V. PTR (SACCESS3(FATHER)S) = SON*	
*HFITE3 = TRUE \$	
*BACK*V(SACCESS3(SON) \$) = *	
ATHER S	
CK H (SACCESS3 (SON) S) =	
ALTOTO TA CCESS SISON ST = 0 S	
ETTER # TOLE &	
ETURN \$	

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANMER Design diagram of Dashes

MATERIA TO THE TOTAL TO THE TOTAL TO THE TOTAL TOTAL TOTAL TO THE TOTAL	COURT IN THE STATE OF THE STATE	
ME PROCEDURE DASHES  FULS A STRING ON  UNIO COL OF PUTOUT RECORD  UNITEGER \$  COLUMN ON WHICH DASHES *  COLUMN ON WHICH DASHES *  COLUMN ON WHICH OF OUTPI).  UT RECORD ON WHICH  SAFEARY  SAFEA		
HE PROCEDURE DASHES  GITH LINGTH, STRING ON  WALUE OF OUTPT).  LINTEGER \$  COLUMN ON WHICH DASHES  WITH INTEGER \$  STRING TO BE	· 医生物 (1) 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
THE STATING ON DASHES OF GITH LINGTH, STATING ON THAT COL. OF PUTDUT RECORD  OR TEHPC (DEPENDING ON VALUE OF QUIPT).  INE INTEGER \$  COLUMN ON MICH DASHES *  OF THE STATING TO BE *  STATING TO THE *  STATING	THE PROCEDURE	
UNN COLL OF PUTOUT RECORD  RALUE OF OUTPIT.  OL INTEGER 8  SINE INTEGER 8  SIN		
VALUE OF OUTPT).  OL INTEGER \$  OLLUMN ON WHICH DASHES  INE INTEGER \$  INE INTEGE	PUTOUT RECORD	
DL INTEGER 8  COLUNN ON WHICH DASHES **  INE INTEGER 8  SAPEAR.**  SAPEAR.*	CDEPENDING ON	
OL INTEGER \$ COLUMN ON MHICH DASHES * COLUMN ON MHICH DASHES * INE INTEGER \$ CIT PECORO ON MHICH S APPEAR* CIT OF DASH STRING TO BE* CIT OF TO TEMP OR OUT LINE * COLUMN ON MHICH \$ COLUMN ON MHICH S APPEAR* CIT OF TEMP OR OUT LINE * COLUMN ON MHICH S COLUMN ON THE STRING TO SERVICE (\$5.0 LNGTH\$) (OUT LINE \$ COLUMN ON MHICH S COLUMN ON THE SERVICE (\$5.0 LNGTH\$) (OUT LINE \$ COLUMN ON MHICH S COLUMN ON THE SERVICE (\$5.0 LNGTH\$) (OUT LINE \$ COLUMN ON MHICH S COLUMN ON THE SERVICE (\$5.0 LNGTH\$) (OUT LINE \$ COLUMN ON MHICH S COLUMN ON THE SERVICE SA COLUMN ON MHICH S COLUMN ON THE SERVICE SA COLUMN ON MHICH S COLUMN ON THE SERVICE SA COLUMN ON MHICH S COLUMN ON THE SERVICE SA COLUMN ON MHICH S COLUMN ON THE SERVICE SA COLUMN ON MHICH S COLUMN ON THE SERVICE SA COLUMN ON MHICH S COLUMN ON THE SERVICE SA COLUMN ON THE SERVICE SA COLUMN ON MHICH S COLUMN ON THE SERVICE SA COLUMN ON THE SERVICE		
OL INTEGER \$  COLUMN ON WHICH DASHES *  INE INTEGER \$  INE INTEGER \$  SAPEAR**  AND		
COLUMN ON WHICH DASHES  INE INFEER S  SAPEAR.  NGTH INTEGER S  ASH H 132 P  LIPT B S  PUT TO TEMP OR OUT LINE  BY THE STATEMENT OF STAT	*ITEM COL INTEGER \$	
NGTH INTEGER 8  S. APPEAR.  S.		
UT RECORD ON WHICH  S. APPEAR.  NGTH INTEGER 8  IT IN 132 P  UIPT B 8  PUT TO TEMP OR OUT LINE 8  H. 100 TP 1 8 BYTE (\$COL, LNGTH\$) (DASH) 8 8  H. 100 TP 1 8 BYTE (\$COL, LNGTH\$) (OUT LINE (\$COL, LNGTH\$) (DASH) 8 8  H. 100 TP 1 8 BYTE (\$COL, LNGTH\$) (OUT LINE (\$C	TINE THEFEE	
ASH H 132 P  ASH H		
NGTH INFGER 8  STH 05 DASH STRING TO BE  ASH 132 P  LIPT B 8  PUT TO TEMP OR OUT "LINE **  *********************************		
UPT B \$  UT TO TEMP OR OUT LINE *  **********************************	A LNGTH INTEGER \$	
UTPT 8 \$ PUT TO TEMP OR OUT LINE ***  ********************************	LENGTH OF DASH STRING TO	
UTPT 8 \$ PUT TO TEMP OR OUT LINE ** **********************************	H 132	
UTPT B \$  PUT TO TEMP OR OUT LINE *  ***-*******************************		•
UIPT B \$ PUT TO TEMP OR OUT LINEOUTPT \$BYTE(\$COL, LNGTH\$)(DASH) \$CATF (\$CATF (\$		
UJPT 8 \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$		
PUT TO TEMP OR OUT LINE **  *********************************	(	
	IN OUTPT 8 \$	
**************************************	***************************************	
**************************************	THE RESIDENCE OF THE PROPERTY	
***-**OUTPT & ****BYTE(\$COL, LNGTH\$)(TEMPC) = * **********************************	***********	*****
+-ORIF 1 \$BYTE(\$0, LNGTH\$)(DASH) \$	*-+-*OUTPT \$ **BYTE(\$COL.	
+-CRIF 1 \$BYTE(SCOL, LNGTHS) COUT-LINE(SA 	essesses e BYTE(\$0, L	
+-*CRIF 1 \$ *BYTE(\$COL, LNGTH\$)(OUT*LINE(\$A ************************************		*******
ecterates (CESS'OUT(LIME)S)) = BYTE(30, CLNGTHS) (OASH) S ecterates ecterates extra	1 \$ *BYTE (\$COL.	*LINE (SA
Company of the control of the Contro	*CCESS*OUT(LINE) \$)) =	SYTE( \$0.
::	サーチャルカラカマル・ナータスプ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	*******
-	S LUSN S	

\* DISPLAYSTUMPREF CREATES AND CUTPUTS A STUMP REFERENCE COUTPUTS A STUMP REFERENCE HORIZONTALLY OR VERTICALLY FROM THE GURRENT DIAGRAM CODE BLOCK, THE BOX CONTAINS THE REFERENCE NUMBER OF THE STUMP HICH IS OBTAINED FROM THE PAGE REF ITEM OF THE STUMP'S GROUP ENTRY. \*PROC DISPLAY\* STUMP\*REF (HORIZ) \* \*ITEM TOP INTEGER S \* \* \*ITEM BOTTOM INTEGER \$
\*\*\* THE BOTTOM LINE OF THE
\*\* DISPLAY BOX\*\*
\*\*\* TIEM DISPLAY INTEGER \$
\*\*\* APPEARS\*\*
\*\* TIEM HOAZZ B \$
\*\*\* TYPE OF DISPLAY, HORIZ OR
\*\* VERT.\*\*
\*\* TIEM INDEX INTEGER \$ \* \*ITEM LNGTH INTEGER \$ .. THE PROCEDURE \* IF DE BUG13 \$ . \*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*

\* C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of display stump ref \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* S \* OUT (MESSAGE, MESS'SW) \$ \* \*REMQUO(BOTTOM, PAGE"LNGTH =\*
\* PAGE, LINE) \$
\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*

\* PAGE, LINE) &

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design Diagram of Display-Stump-Ref

FETH	* IF LINE LS HEADROOM &	H SDISPLAY = PAGE * PAGE LNGTH +*
F3 REC(SINDEXE) AND INDEX LS GROUP-AVAIL) F3 REC(SINDEXE) AND INDEX LS F3 REC(SINDEXE) AND INDEX LS GROUP-AVAIL) STUND: STUND: STUND: STUND: STUND: STUND: TE GROUP STORE WITH	*INDEX # CUR'GROUP S #	
F3"REC(SINDEXS) AND INDEX LS GROUP-AVAIL 1 DO WHILE ((/ H) DO WHILE ((/ V'SON /) NQ F3"REC(SINDEXS) AND INDEX LS GROUP-AVAIL 1 STUMP) 1 S STUMP) 1 S STUMP) 2 STUMP) 1 S STUMP) 2 S STUMP) 1 S STUMP)	REFERRING TO THIS STUMP	
GROUP AVALLS  F3 REC(SINDEXS) AND INDEX LS  GROUP AVALLS  STUMP) S  STUMP) S  UT(***  TE GROUP STORE WITH	"DO MHILE ((/ H°SON /) NQ • F3°REC(SINDEXS) AND INDEX LS • GROUP-AVAIL! DO WHILE ((/ H°	secretations NEXT (SINDEXS) & *
TARS (LEFT START, 6. TOP) \$  YTE (SLEFT START, 6. TOP) \$  AY15) = 64(° ° ° ° ° ° ° ° ° ° ° ° ° ° ° ° ° ° °	- 3 - 3	
(\$1) \$ IFORMAL(PAGE*REF(\$INDE* (\$1) \$ NGTH = LENGTH(TEMPC) \$ ** NGTH = LENGTH(TEMPC) \$ ** NGTH = LENGTH(TEMPC) \$ ** LNGTH\$1(IEMPC) \$ ** LNGTH\$1(IE	STARS (LEFT START, 6, TOP) \$ SYTE(SLEFT START, 6, TOP) \$ 68) (CUT LINE (SACCESS OUT (DISP AY) \$) ) = 64 (*) \$	
LNGTHS) LEWTE(\$6.  LNGTHS) (TEMPC) \$ ***********************************	EMPC = IFORMAT(PAGE REF(SINDE .  (NGT) = LENGTH(TEMPC) \$  YTE(SLEFT START + 2	
	LNGIHS) 1001 - LINE (SACESS OUT (* )ISPLAY) 1) = 8YTE(\$6,	
	J.	

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAN OF DISPLAY"STUMP" REF

\*FROM PAGE (SINDEXS) = PAGE NO S. \*

\*RETURN S .

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF DOTS

						<u> </u>	:																					
•	DOTS (COL. LNGTH. LINE) \$	8		-		=	INE TH RECORD OF PUTOUT		•	•	•	•	•	•	•	-	•	•	•	•	•	•	•	4	= 8YTE (\$0. *	•	•	
	bit.	9		ā		4			:							BE OUTPUT							"	2				
:	2			5		0	5		:				5			5			:	:	:	:	-	¥	3			
:	3			C	=	4	2		:				0			0			•	•	•			H	W	-		
		2		-	ă	2	5				·		PUTOUT			w			:	:			:	5	2			
:				5	1	z	•		:		Z								:	:	:			5	8	186		
:	=:			0	1	E	4		:		2		THE			2			:	:	•	:		ō	"			
:	9	88		w	Ž	2			***********		COLUMN		=		0	UMBER OF DOTS TO				•				-	-			
				5		2	2			~	٥	ER			3	13			:	:	:		:	H	2			
:	.:			6		-	8		:	3	2	9	0		3	8		2	:	:	:	:	:	-	3	•		
:	2:			Ö	0	4	W			W	Ē	Ξ	œ		Z			13				•		ž	Z	-		
:	3:	33		8	9	0	_			Z	a	Ä	8			0		-		:	:	:	:	-	=	5		
:	5			•	Z	Z	Ξ		:	-	Z		=		Ξ	~		I	:	:	:	:			=	9		
:	5:			W	2	=			:	×	S	2	Z	•	9	W		=						ō	3	=	*	١
•	8			=	S	4	ž			ິບ	w	3	w	ORO	3	I		TOO	-	•				*		Ï	2	ċ
:	0:					5	3		****	-	THE STARTING COLU		I	8	-	3		I	Z	:	:	:		3	SS	5	2	:
:	2:								:	W		ITEM		H	TEM		:	TEM	13246	:	:			E	W	=	FTURN &	١
•	.PROC									=		=	:	-	Ξ	:		=	-	:	:	:	:	6	ü	· LNSTH\$) (DOT) \$	30.	į
•		15							•	•		•	•	•				•	•	•	•		•	•	•		•	

\*

```
TOX. TOY.) S

TOX. TOY.) S

TOX. TOY.) S

TOX. TOY.) S

TOX. TOY. THE PROCEDURE DRAWLINE

DRAYS A LINE FROM THE POINT

HAVING X ADO Y CORDOLIASIS

(IN TERRY OF CORDOLIASIS

(IN TERRY OF CORDOLIASIS

(IN TERRY OF TOY.) THE FROM Y TO THE

POINT TOY. THE FER S

TOX. TOX. TOY. TOY. S

TOX. TOX. TOY. S

TOX. TOX. TOX. S

TOX. TOX. S

TOX. TOX. S

TOX. TOX. S

T
```

C S DRAFER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRAMER DESIGN DIAGRAM OF DRAM'LINE

*OUT(MESSAGE, MESS*SA) &	**************************************
NG TO'Y S. NG TO'X AND FRON'Y	LINES "  LINES "  LINES "  GYTE(\$TO Xx, 1\$) (OUT 'LINE(\$ACCE*  *SSOUT(FRONT 'S IS) " LIN(*) \$  *TO UT (FRONT 'S IS) " LIN(*) "
1F (FRON'Y EQ TO'Y DR TWO'LINES) AND TO'X - FRON'X - GR 1 S	*** A HORIZONTAL **  *********************************
IF (FROM'X EQ TO'X OR TWO'LINES) AND TO'Y - FROM'Y - GR L 1	FOR I = FROM V + 1, 1, TO Y = 0 000000000 0000000000000000000000
THE DEBUGS CONTRACTOR OF THE PROPERTY OF THE P	**************************************

\*RETURN S

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER DESIGN DIAGRAM OF DRAM"LINE

72 FROM 71 \*\*
\*\* 72 FROM 71 \*\*
\*\* FROM 71 \*\*
\*\* FROM 061; PAGE\*\*
\*\* LINE ) \$

A District No. of

CORREST TWINSTADM TOKEN ACEDIATES OF COMPANY OF STREET

						ACCE.		
DESIGN DIAGRAM OF DRAW LINE					************************	*BYTE (\$10"X. 18) COUT"LINE (\$ACCE"	\$ 1.31	***************************************
SAH OF					•	18)	= =	
0146		FROM	****		****	STO.X	1(1)\$	
DESIGN	***************************************	. 73 FROM 71	**************		:	·BYTE	10.SS.	•

\* 55 2839 or 7

SWARDARD TO DEARLY BEARING THE STRUCTURED DESCRIPTION OF THE STRUCTURE OF

C S DRAFER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF EXTRACT NAME

** THE FUNCTION EXTRAGT*NAME ** EXTRACTS AND RETURNS THE NAME	TA CASA A
EXTRACTS AND RETURNS THE NAME	
. OF A MODILE MEAN	
* 10000	100 mm m
TTEM DONE B \$	
TITEM EXTRACT NAME CHARACTERS	TO SEE THE PROPERTY OF THE PRO
*** RETURNS THE MODULE NAME*** *********************************	
TIEM INDEX! INTEGER 6	
TEN INDEX FOR STATE OF STATE O	
*ITEM LNGTH INTEGER \$	
* STRINGS.**	And the contract of the contra
STORES FILE 2 RECORD TEXT	
*JONE = FALSE \$	
* SUGGES OF CHARACTER STREET OF	
	SSOURCE LINE = F2"LINE (\$4 CCESS2" "(FILE2"INDEX + INDEX"F2) \$ " "INDEX = 6 \$ " "INDEX = 6 \$ "
のであるから、 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	TO COMPANY OF THE PARTY OF THE
· 中国の大学を大学を表現して、	
	*DO WHILE (NOT DONE AND INDEX - * *****
* Charles	***************************************
	A 100 C C C C C C C C C C C C C C C C C C
	*****************
	SPORTS INDEX - 2 & CONTROL OF SPORTS

C S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF EXTRACT NAME

	** MAIN PGM HEAD **  ** EXTRACT* NAME = PGM* NAME \$  ** FXTRACT* NAME = CNVERT(EXTRACT***********************************	ORIF 1 S	* 18) (SOURCE LINE) EQ 1H() AND ***********************************	ADVANCE BEYOND SPACES	**************************************	INDEX1 = INDEX \$ SAVE STARTING POINT	* 18) (SDURCE*LINE) NO 1H() AND * *********************************	TRUNCATE NAMES SPANNING FILEZ RECORDS	
--	--	----------	---	-----------------------	--	---------------------------------------	---	---------------------------------------	--

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER DESIGN DIAGRAM OF EXTRACT NAME

 STEENSTREET BUTCH TRACT NAME OF THE CONTRACT O
*HESSAGE * CATCHESSAGE,  • EXTRACT NAME; \$  •OUTCHESSAGE, HESS SM) \$

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAHYER Design Diagram of Extract\*Name

. 77 FROM 74 .

"PROC"." "PROC"." "PROC"." "PROC"." "PROC"." "PROC"." "A\$) (SOURCE"LINE) EQ 4H(PROC) "	CHECK FOR "CLOSE"	S\$)(SOURCE"LINE) EQ 5H(CLOSE) ***** GOT IT **  * \$\$)(SOURCE"LINE) EQ 5H(CLOSE) ************************************	•
TINDEX - 1 LQ LNGTH &	LHGTH S THOU THOEX - 2 LQ **		***************************************

THE FUNCTION EXTRACT TEXT

CORDANCE MITH THE LINE

BREAK INFORMATION CONTAINED

IN THE SET OF FILE 4 RECORDS

HICH CORRESPOND TO THE

STATEMENT UNIT.

TEM BYTE PTR INTEGER \$

"ITEM BYTE PTR INTEGER \$

"ITEM BYTE PTR INTEGER \$

"ITEM EXTRACT TEXT CHARACTER \$

"CONTAINS TEXT TO BE

RETURNED.

"ITEM FIELDI INTEGER \$

"TEM FIELDI INTEGER \$

"TEM

\*\* LINE PTR IS THE FILE 2
\*\* RECORD WHICH CONTAINS\*\*
\*\* THE LAST BYTE TO BE
\*\* RETURNED\*\*

C S DRAPER LABORATORY JOVIAL STRUCTUREU DESIGN DIAGRAMMER DESIGN DIAGRAM OF EXTRACT\* TEXT

* ************************************	A ST. SON OC. I FALSE SA		**************************************	*****	**********  **********  **********  ****	* S * S * S * S * S * S * S * S * S * S	**************************************
		CRIF 18TEXT-LINE 3 FIELO2) 5) 5	HIII				### ### ### ### ### ### ### ### ### ##

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER

T. FINE. F	FIELD2 = LINES OUT(SF4 PTRS) S.
	.14.
	1
IF UE BUGIE & GALLENESSAGE	THESSAGE = 7H(EXTDACT) &
•	= IFORMAT(F4.PTR) &
**************************************	*MESSAGE = CAT (MESSAGE, T.MESS) *
•	
	*T*MESS = 1H(=) \$
. *OUT CHES	*OUT (MESSAGE, MESS*SW) \$
. * HESSAGE	= IFORMAT(FIELDI) \$ +
. *MESSAGE	*HESSAGE = CATIT MESS, MESSAGE)*
•	
. OUTCHES	SAGE. MESS'SH) \$
. *HESSAGE	= IFORMAT(FIELD2) \$ .
. * HESSAGE	*HESSAGE = CAT(I"MESS, MESSAGE)*
* *OUT CHES	*OUT THE SSAGE, MESS'SWI &
. * HESSAGE	= IFORMAT(FIELD3) \$ *
. * MESSAGE	= CAT(I"MESS, MESSAGE) .
•••	
. *OUT (MES	*OUT (MESSAGE, MESS'SH) \$
******************************	:

FIFEITH \*-+-\*FIELDZ EQ 1 8 \*----

\*EXTRACT\*TEXT = TEXT\*LINE \$ \*

.. ONLY 1 LINE ..

**************************************	Tr ucedule s				 *RETURN S .	
	(EXTRACT TEXT . MESS"	*******				
121 12	SW) S .		Trans.	The same	Mark W	
			No. of the last	0.000.00	を 10 mm 1	0.000
			大学 かんとう	1000	大 丁 二 日 年 日	
				を おおかける		
			1000		9 46 0	

\* \*\*\*\*\*\*\*\*\*\*\*\* \*TEXT"LINE = F2"LINE (\$ACCESS2(L\* . . IT DOES .. \*\* \* 1 ANG LNGTH EQ MAXCOL \$ \*\* NEXTF4 DOESN'T PERTAINS TO THE STAT UNIT \*\* \*CHOTH = LENGTH(TEXT\*LINE) \$ \*EXTACT\*IZXT = SUBSTR(TEXT\*LIN\*
\*E\* FIELD3, LNGTH - FIELD3 + 1)\*
\* \* \*\* SEE IF STHE UNIT CONTINUES TO NEXT F2 REC \*\* \* \* \*\*\*\*\*\* \*LINE PTR = F2 REC(SNEXT F48) S\* \* \* .. ANOTHER F4 REC FOLLOWS .. \*BYTE \* FZ BYTE (\$NEXT \* F4\$) C S DRAPER LABORATORY JOYLAL STRUCTURED DESIGN DIAGRAMMER Design diagram of extract-text \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\* ---ORIF 1 \$ -----\*\*\*\*\*\*\*\*\*\* . 82 FROM 80 . \*\*\*\*\*\*\*\*

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF GENERATE CONTENTS HEADER

TERPC = IFORHATIPAGE NO) & ** **LNGTH = LENGTH(TEMPC) & ** **SYTE(SPACE BYTE, ** **LNGTHS (TEMPC) & ** **ENTERSON & ** ** **ENTERSON & ** ** **ENTERSON & ** ** **ENTERSON & ** ** ** **ENTERSON & ** ** ** ** ** ** ** ** ** ** ** ** **
---

BYTE(SMARGIN, 178) (OUT-LINE(SA- "CCESS-OUT(LINE'NO)S1) = "BYTE(S6, 178) (TEMPC) S "LINE'NO = LINE'NO + 2 S "RETURN S		. ( \$ V .	•			•	****
ARGIN, 178) (C 16, 178) (TEMPC 18, 178) (TEMPC 18, 178) (TEMPC 18, 178)	を 日本 七 中 田 田	DIT-LINE	* -	* ::	8 2		******
ARGIN. DUTCLIN 16. 178	●次万 日子のの	178) (0	E. NO) \$)	1 (TEMPC	E.NO .		******
	中華のの記録	TARGIN.	DUTCLIN	16. 178	NIT =	*RETURN \$	******

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF GET-FI-REC

Charles .	
**************************************	
. TOTEC TO DEAD THE NEXT FIFE	
* 1 RECORD. IF AN END OF FILE	
. IS ENCOUNTERED, IT SETS	
. EDFILE TO TRUE	
STEM E. DID TUTEGED .	
O. THE NUMBER OF THE FILE 1 *	
*LAST*F1 = LAST*F1 + 1 \$ *	
*F1*PTR = ACCESSI(LAST*F1) \$ *	
***************************************	
1	
South was likely before you grant and give a part of an	
BUG2 \$ .	******
	FILE EOF) S
Press, plan Endines	OUT (MESSAGE, MESS SW) &
• RETURN S •	
*FILED************************************	
a company of the comp	
*STHT" TOKEN = FIELD2(SF1"PTRS) "	
- Carlo Constitution of the Constitution of th	
*F2*RECS # FIELD3(\$F1*PTR\$) \$ *	
*RETURN S	
**************************	

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of Iformat

THE FUNCTION IFORMAT  ACCEPTS AN INTESER, AND RETURNS A CHARACTER STRING (IN CONVERTED FORM)  REPRESENTING THE INTEGER.*  THE INTEGER S CONVERTED* THE STRING TO BE RETURNED*  THE STRING TO BE RETURN		PID	ER.		NED	: 3	• •
CHAM A TO T C	14	IFORMAT ESER. AN ACTER ST FORM)	ME INTEG	<b>.</b> #	ACTER S BE RETUR	R ENCODE	LF ORMAT)
CHAM A TO T C	T N	AN INTER	TING T	TEGER :	NG TO	ARY FOR	6 S NVERTCI
OS COA THENTH MAN AND AND	IF ORM	THE FU CCEPTS ETURNS IN CONV	EPRESEN	HE INTE	THE STRI	TTEM M6 H 6	MAT = H

CONTRACTOR OF THE PROPERTY CONTRACTOR OF CLOSE

\*PROC INIT'BLOCK"CONSTANTS & \*

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* • THE PROCEDURE
• INIT'BLOCK'CONSTANTS IS
• CALLED BY PART2 TO INITIALIZE
• THE VARIABLES WHICH WILL BE
• CONSTANT THROUGHOUT THE
• CREATION OF THE CODE BLOCK. \*HF4\*SIOP = F4\*END(SINDEXS) \$
\*V\*SON = V\*PTR(SINDEXS) \$
\*V\*SON = V\*PTR(SINDEXS) \$
\*V\*FATHER = BACK\*V(SINDEXS) \$
\*V\*FATHER = BACK\*V(SINDEXS) \$
\*PAST\*HIND = F4LSE \$
\*\*PAST\*HIND = \* \*BCTTOM\*LINE = STOP\*LINE(SINDEX\* \*TOP-LINE = START-LINE(SINDEXS) \*LEFT\*COL = START\*COL(SINDEXS) \*RIGHT COL = STOP COL (SINDEXS) \*F4.START = F4.BEGIN(SINDEXS) \*WIDTH = BLOCK WIDTH(SINDEXS) \* + OISP \$
\*MIDPOINT = MIDPT(\$INDEX\$) +
\* DISP \$ . ......... \*\*\*\*\*\*\*\*\*\*\*\*\*\*

IGN DIAGRAMER	INE IN	ACCESS C.		
VIAL STRUCTURED DESIC LOCK CONSTANTS	CASE OF BUFFER RUNC	**************************************		STATE AND A
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design Diagram of Int'block" Constants ************************************	TE V'SON GR 0 S ****** GET V'SON'S TOP LINE IN CASE OF BUFFER RUNOVER ***		**************************************	

AD-A052 733 CHARLES STARK DRAPER LAB INC CAMBRIDGE MA F/6 9/2 JOVIAL STRUCTURED DESIGN DIAGRAMMER (JSDD). VOLUME III. PROGRAM--ETC(U) FEB 78 G GODDARD, M WHITWORTH, E STROVINK F30602-76-C-0408 R-1120-VOL-3-PT-3 RADC-TR-78-9-VOL-3-PT-3 NL UNCLASSIFIED 2 OF 3 AD AO 52 733 THE PARTY

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANMER DESIGN DIAGRAM OF INITIALIZE

	OCEDURE INITIALIZE	EXECUTION	*************	NOT	•	= HEADROOM \$	ST* MAX / 2 8	ST**** + 10 8	PAGE WIDTH - 1 8 .	6
PROC INITIALIZE S			****************	SINGLE SPACE =	DOUBLE SPACE &	"LAST" LINE = HE	LOW-LIM = ST. M	"MAX" WIDTH = ST	PAGE" WIUTH = P	DPEN INPUT FILES

\*PROC INITIATE\*RECORD(INIT\*REC) \*

\*

E. REC A CORD.		" " "	St. CF.
RE INITIATE REC VALUES IN A FILE 3 RECORD.	INTESER & BLOCK BEING	4 "END(\$ACCESS3(INIT"REC)\$) = LAST"4 \$ INES(\$ACCESS3(INIT"REC)\$) = LINES"OUT(\$ACCESS4(FIRST'4)\$) \$	*BLOCK*WIOTH(\$ACCESS3(INIT*REC)**) = MAX*LINE*LNGTH(\$ACCESS4(F*IRST*4) \$\) *TRST*4) \$\) *TRST*4) \$\) *TRST*4) \$\) *TRST*4) \$\) *TRUE \$\) *WRITE3 = TRUE \$\) *RETURN \$\)
DURE I NG VAL ED FIL	INTESER BLOCK BE	GESSAC	CCESS3 LNGTHI ESS3(I
* THE PROCEDURE SETS STARTING VA NEMLY CREATED FI	. n. o. o. %	A CCESS CCESS3 UT (\$ A C	TOTH(SACCINE LNS) S S S S S S S S S S S S S S S S S S
SE TS NE HLY	REC NO.	LAST .4 LAST .4 INES (\$A LINES O	STAT-UNI STAT-UNI STAT-UNI BENEW RETURN &
		:: <u>:</u> :::	# ST

F3 INDEX) \$  F3 INDEX) \$  F3 INDEX) \$  F4 INDEX	
T LINKS UP TABLE EING	
r LINKS UP TABLE EING	を受ける。 また できた できた できた できた できた できた できた できた できた でき
	· 一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个
	在於在衛門衛衛衛衛門不然發展的不過之間被軍衛衛衛衛門以及軍事
	and the state of t
TOR A TREE MALEN TO DE LAG	
INITIALED	
	· 医电子性 医甲状腺 医甲状腺 医甲状腺 医甲状腺 医甲状腺 医甲状腺 医甲状腺 医甲状腺
	SPECIAL SECTIONS
	TO ADDRESS TO BE ASSESSED TO A PROPERTY OF THE PROPERTY OF THE PARTY O
***INDEX OF FILE 3 RECORD BEINGS	· 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
*ITEM GROUP-TYPE INTEGER \$ .	(2) 日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日
. TYPE IS MODULE HEAD OR STUMP.	· · · · · · · · · · · · · · · · · · ·
*******************	* 1500 m 4500 km 810 km 810 km 100 km
	A CARD CLASSICAL STORY OF THE S
•	
THIS PROC INSERTS THE NEW	
. GROUP INTO THE GROUP	
	ANNESSES.
	the second of
DECLARATION	
THE GROUP STORE IS A LINKED	
APPROPRIATELY FOR OUTPUT	1000 1000 1000 1000 1000 1000 1000 100
• PROCE SSING. •	The second of th
• ALTHOUGH UNECESSARY FOR	高度中央的第三人称单数 医阿拉克氏试验检尿病 化二甲酚 电电子电 化电影电影
. "COULD BE ACCOMODATED BY	
* INTRODUCING A GROUP	2000
. ENVIRONMENT	我都是我们的情况的情况不是不多的的时候。 网络阿拉拉里亚
. STACK	100 公司 600 100 100 100 100 100 100 100 100 100
*************************	***************************************
*IFEITH *-+-*GROUP-AVAIL GR GROUP-MAX S *-	*HESSAGE = 20H(GROUP STORE*
•	*PHZERR(MESSAGE) \$
	.!
	* 4
SETTING GEORGEM OL INSENT	FROM A PROC

58.88 36

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF INSERT

**************************************	erruserruserruserruserruserruser (ROLD) = CROUP'AVAIL S				AVAILS) =		
errectes errectesrates electroristes	SANGE OF THE STANGE OF THE STA	**************************************	**************************************	#NEXT(\$GROUP.AVALLS) = +  * NEXT(\$LAST*STUMPS) & +  *NEXT(\$LAST*STUMPS) = +  * GROUP.AVALL & +  * GROUP.AVALL & +  * LAST*STUMP = GROUP.AVALL & +  **********************************	**************************************	#ORIF 1 \$ ***** ****************************	# GROUP = GROUP AVAIL \$ # F3 REC(&CROUP AVAIL \$ # F3 INDEX \$ # F4 INDE
PILE I CALL TO THE PILE TO THE		A SOUR SECURITY AND SECURITY OF SECURITY SECURIT			FEITH	ORIT	Elizabeth a control of the control o

C S DRAFER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of Legal'Stump

PPROC LEGAL'STUMP(STMT-TYPE) & •

			OR * **********************************	*LEGAL*STUMP = TRUE \$ *
	** THE FUNCTION LEGAL STUMP ACCEPTS A CODE BLOCK TYPE AS INPUT AND CUTPUTS A I IF THE CODE BLOCK IS A LEGAL STUMP ROOT. OTHERISE, IT RETURNS A 0. **	TIEM LEGAL'STUMP B \$ """ HOLDS THE RETUKN VALUE" " THE STATITYPE INTER \$ "" THE INPUT CODE BLOCK TYPE" "	**************************************	ORIF 18
•	ACCEPTS INPUT AN CODE BLC	TEM LEGAL STUMP B S THOUS THE RETURN ITEM STATT THE RETURN ITEM STATT THE INTEG		PRETURN S

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANMER DESIGN DIAGRAM OF MAX

THE THE TREE TO THE TENT TO TH	PROC MAXINTI, INT) S THE FUNCTION MAX RETURNS THE LARGER OF THE TWO VALUES PASSED TO IT	FITTH INIT INTEGER 8  FITTH INIT INTEGER 8  FITTH INT INTEGER 8  FITTH HAX INTEGER 8	THE STATE OF INTERSTREET OF STATE OF ST
--	---	---	--

	-35					
					10.	
	- 35		VR.		18	36
	TAL.					
	*					
	: E					
	- 70				100	
			38	-	の日本の日本の日本の日本の日本	
			36	44	546	
				=		
			2	-		
			-	_		
			œ	-		
	31/80		3	>		
	-32		=	00	- 60	
	の 日本の 日本の このは 日本の	10年の日 10日 10日 10日 10日 10日 10日 10日 10日 10日 10	-	_		日本社 ののにないないののののののののはないの
			*	2		
			Œ	-		
			- 20	-		
			z		100	
	780		-	ш		
	-36		-	=		
			. 17	=	-	
			-	-	•	
			-			23
	-39		0			
	- 10		-	0	-	
			-		-	
	102			~		
	80		=	-	-	
= = :			=	w	_	
		-95	9	3	-	
		35.		v		
		-46		w	0	
			W		-	
			Ŧ	-	7	
			=		14	
			100	=	~	
oppoc minimit, into s	Act 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.			. THE LESSER OF THE THO VALUES	-	
. 0.	3	194		-		
			-			_
	1		-	-		
			- 2			
	18					
	3					

	TE + NEW ENGINEER CA. AND	· 日本祖祖の公司者所以上各名の以下の公司				· · · · · · · · · · · · · · · · · · ·			********	IT 8 BIN = II	
*ITEM INT. INTEGER 8 .	INPUT PARAMETER"	*ITEM INTZ INTEGER 8 .	INPUT PARAMETER"	*ITEM MIN INTEGER \$ .	*** THE RETURN VALUE"	***************************************	A. J. SHOP TRUE OF CHARLESTEE	A LINE AND A DESTRUCTION OF THE PARTY OF THE	*************	" IFEITH " INT LS INT & " ON IN E IN	

S S TINI H NIME S T LINOS .

\*RETURN S \*

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of Output\*Box\*Bottom

AGE ON WHICH THE BOX  AGE ON WHICH THE BOX  AGE ON WHICH V'SON'S	BOTFOT BCX BOTTON OUTPOTS THE BOTTON OF THE COOK. IT  * ALSO PERFORMS SOME DOUBLE  * BUFFERING CPTIMIZATION. **	
HA INTEGER S  "USED IN CALLING  "INTEGER S  INTEGER S  INTEGER S  INTEGER S  INTEGER S  LINE OF CONNECTING LINE  "INTEGER S  LINE OF CONNECTING LINE  "INTEGER S  LINE OF CONNECTING  "INTEGER S  LINE OF CONNECTING  "INTEGER S  "INTEGER	*ITEM QUOI INTEGER \$  ***THE PAGE ON MHICH THE BOX *  **BOTTOM APPEARS**  *ITEM QUOZ INTEGER \$  ***THE PAGE ON WHICH V*SON*S *	
INTEGER S  INTEGER S  LINE OF CONNECTOR S  LINE	* BOX TOP APPEARS**  *ITEM REM1 INTEGER \$  * REM2UO**  *ITEM REM2 INTEGER \$  ***********************************	
169 S * * * * * * * * * * * * * * * * * *	TIEM XI INTEGER \$  "COLUNN OF CONNECTING LINE" •  "ITEM Y1 INTEGER \$  "ITEM Y2 INTEGER \$  "" END LINE OF CONNECTOR" •  "" END LINE OF CONNECTOR" •	
######################################		
	SIMITYPE EQ CONNENT'S &	YF(&LEFT*GOL + 1, 4) COUT*LINE(\$ACCESS*OUT(BOTT** H*LINE)\$) = 1H(*) \$

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANMER Design Diagram of Output-Box-Bottom

\$ 1 J. I Y I Y I Y I Y I Y I Y I Y I Y I Y I	S *STARS(LEFF.COL, MIDTH + 2.*  **********************************
LINE NO BOTTOM-LINE S + -LAST-LINE = MAX(LAST-LINE, + - LINE NO) & AX	
NUMBER OF INVOCATIONS OF ** TRANSFER PRITE OUT BY ** DRAWING PART OF ** ** CONNECTING LINE NOW IF	UCE THE TIONS OF OUT BY E NOW IF A
	AEMQUOLLINE-NO, QUI'BUF-SIZE == * QUO1, REM1) % *REMQUO(SON'TOP, QUI'BUF-SIZE == * QUO2, REM2) %
	######################################
	* * * * * * * * * * * * * * * * * * *
	* OUT (MESSAGE, MESS'SW) & * COUTCHESSAGE, MESS'
••••	A BUFFER SPAN DOES OCCUR
07 - N. W.	STREET STREETS

***************************************
*Y2 = (QUO1 + 1) * OUT*BUF*SIZE*
errorsessessesses essessessessessessessessess
**************************************

RETURN S

C S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF OUTPUT-BOX-TOP

OUTPUST HE TROCEDURE OUTPUT BOX TOP  OUTPUST HE TOP LINE  OF BOX NO MED TO HANDLE  HEADINGS HERE  THE OF BOX NO MED TO HANDLE  HEADINGS HERE  THE OF BOX NO MED TO HANDLE  HEADINGS HERE  THE OUTPUST HER	UT * BOX * TO P
THITTYPE EQ CONTROL S STATE STATE EQ CONTROL S STATE S	4E TOP LINE To mandle
**************************************	
ORIF 1 S 1 S O O O O O O O O O O O O O O O O O O	**************************************
++++++++++++++++++++++++++++++++++++++	
**************************************	
: :	### ##################################
	: :
++*ORIF 1 \$ **SIARS(LEFT*COL, MIOTH + 2.*  **********************************	

```
************
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
Design Diagram of Output"Header
                                                                                                                                                                                                                                                                                                                                                                                                                    *OUT(MESSAGE, MESS*SH) S
                                                                                                                                                           OUTPOTS THE PAGE HEADING STARTING ON THE LINE PASSED TO IT.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        *IF TEMP*LINE EQ SPACES(MAXCOL) *
                                                    **********************
                                                                *PROC OUTPUT . HEADER (PAGE . TOP) $
                                                                                                        **********************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 *******************************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           *TEMP*LINE = OUT*LINE($ACCESS*O*
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         *TEMP*LINE = CNVERT(TEMP*LINE) .
                                                                                                                                                                                                                                                     ** STORES STRING LENGTHS**
** TEM PAGE TOP INTEGER $

** THE LINE ON WHICH THE **
** THE LINE ON WHICH THE **
** THE LINE CHARACTER $

** STRINGS**
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             ******************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  ********************
                                                                                                                                                                                                                                          ***********************
                                                                                                                                                                                                                                                                                                                                                                                                                                                .
                                                                                                                                                                                                                                                                                                                                                                                                                                                            *************
                                                                                                                                                                                                                                                                                                                                                                                                                   * IF DE BUG17 $
```

C S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRAM1ER DESIGN DIAGRAM OF OUTPUT MEADER

**TEMPC = BYTE(\$18, ** ** MAXCOL\$)(HEADER(\$18,) \$ ** MAXCOL\$)(HEADER(\$18,) \$ ** ***E(\$18, MAXCOL\$)(OUT*LIN* **E(\$18, MAXCOL\$)(OUT*LIN* ** *******************************	
*FOR I = 0. 1, HEAD'NO \$ ****TENDC = BYTE(\$6, ************************************	eLINE'NO = LINE'NO + HEADROOM S*
	RETURN S

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANMER DESIGN DIAGRAM OF OUTPUT\*TITLE\*PAGE

eproc output TITLE PAGE & *	
• • • • • • • • • • • • • • • • • • •	
OUTPUTS PAGE OUTPUTS THE	
TITEM LINE INTEGER \$	
* PAGE **	
TITEM LNGTH INTEGER 8	
GITER PAGE INTEGER 5	
DUMHY USED IN CALLING	
* REMOUS.*	
SITEM TEMPSLINE CHASACTER & .	
*ITEM TITLE INDEX INTEGER \$ .	
* ARRAY.	
*ITEM TITLE PAGES INTEGER \$ *	
# TITLES.	
***************************************	
*IFEITH *-+-*HEADING \$ **TILE*PAGES = TITLE*NO /*	
* * * * * * * * * * * * * * * * * * *	
**********************	
:	Contract to
POTENTIAL S TO THE STATE OF THE	
***************************************	
A CALL OF THE PARTY OF THE PART	
A CARACTER STATE OF THE STATE O	
PFOR I = 0, 1, TITLE PAGES 8 "" F HEADING 8 "" FEMPC = IFC	IFORMAT(I + 1) & LENGTH(TEMPC) &

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of output\*title\*page

BYTE(ES, LNGTHS) (TEMPC) S * *TEMP*LINE = BYTE(SS, * MAXCOLS) HANGOLS) S * *BYTE(SHARCIN, MAXCOLS) COUT*LIN* *TEMP*LINE S * **LEMP*LINE S * **	CATALOG STANDARD STAN	**************************************	*) \$  **SYTE(SHARCIN, MAXCOLS)(OUT'LINF ** CEACCESS*OUT(LINE*NO)S)) = * ** TEMP*LINE \$  **LINE*NO = LINE*NO + 1 \$  **TITLE*INDEX = TITLE*INDEX + 1 *	* REHQUOLLINE'NO, PAGE'LNGTH = + PAGE, LINE) \$
** BYTE(E6, LNCTHS)(TEMPC) & "TEMP'LINE = BYTE(86, ** MAXCOLS)(MEADER(808)) & "BYTE(3AREGIN, MAXCOLS)(OUT"L ** FEATCHER SOUT(LINE ** TEMPLINE & "TEMPLINE SOUTHER SOU	* PAGE, LINE'S STATEMENT TO PAGE "LNGTH = PAGE "LN	*00 WHILE ILINE GQ START*TITLE *		
				PAGE NO # TILE PAGES + 1 8 *

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of Part2

\*PROC PARTZ S \*

	THE OF		**************************************	March   Marc	**************************************	** TST BLOCK IN CUR'GROUP
•	• • • THE PROCEDURE PART2 IS THE • DRIVER ROUTINE FOR PART2 OF • THE DDG. • •	OTTEM LNGTH INTEGER S  TITEM THAIL INTEGER S  TO AN INTEGER TEMPO  THEN THAIL INTEGER S  THEN THAIL INTEGER S  TO AN INTEGER TEMPO  TO	S TATELET STREET STREETS STREE	**COMPUTE**PAGE**NUMBERS ** **CUR**GROUP = NEXT(\$0\$) \$ ***IST GROUP IN STORE HAIN* ** PGH ***	# CONTRACTOR OF THE CONTRACTOR	

• •• INITIALIZE TRAVERSAL
• STRUCTURE ••

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of Part2

TRAVERSE TOP = 0 S. TRAVERSE STACK(STRAVERSE TOPS) * TA VENE STACK(STRAVERSE TOPS) * TA VENE STACK (STRAVERSE TOPS) * TA VENE STACK	*** SPACES (MAXGOL) S ***********************************	PDO WHILE COURREG GR 0) ***********************************	TRAVERSE TOP = TRAVERSE TOP + 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	FIGURESTHITTYPE NQ 0 S 111.  FIGURE
TRAVERSE TOP TRAVERSE STAR T 0 \$ TADVANCE PAGE			• • • • • • • • • • • • • • • • • • • •	

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of Partz

		原列的學院別為我連合水中では10日本では自己のではなっていることになっ
	•	*******************
	P. P. BONDSONESSES	THE REAL PROPERTY AND ASS.
•••	SE® IF WE NEED A HEADING HERE	
••		
•••	*REMQUO(LINE "NO, PAGE" LNGTH ## # THPI1, THPI2) \$	20 10 10 10 10 10 10 10 10 10 10 10 10 10
••	· · · · · · · · · · · · · · · · · · ·	
•••	*IF TMPI2 EQ 0 AND HEADING \$ ***********************************	**************************************
• •		
•••	*****	
•	COLIPUT BOX BOTTON & +	
•	* 15 15 15 15 15 15 15 15 15 15 15 15 15	
•	•••••	
•••	* 112*	Charles and a few and a second and a few and a
A POLICE DE MINISTER DE PRESENTA DE COMPANION DE COMPANIO		AND AND A CHARLE WAS AND CONTRACTOR
י פי מיינים ארייים ארייים ארייים איינים איינ		
*********************		のできるとのできるとしているとのできるとのできるとのできるとのできるとのできるとしているとのできるとしているとのできるとのできるとのできるとのできるとのできると

SERTOR DESCRIPT STATES TONING SIENTIFIED OFFICE STREET WHEN

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of Partz

NOTA WE WAS TO SEE WAS THE SEC WAS TH		D FROM 108 *				*********************	- LENGTH (PROC" NAME (SCUR"	• GEOUP \$11 S	YTE ( SNAME " BYTE .	LNGT MS) (HEADER (SNAME "INDEXS))	(\$6. LNGTHS) (PROC'NAME (*	CUR. GROUPS) 1 \$	-	\$ (1	NAME BYTE + LNGTH S .	SPACES(TMPI1) 8	HPIZ. THPI1S) (HEADER(SN.	EX5)) = BYTE(\$6.	. THPI1S) (TEMPC) \$	UTPUT HEADER (LINE NO) S	AGE . REF (&CUR GROUPS) =
--	--	--------------	--	--	--	-----------------------	-----------------------------	----------------	----------------------	-----------------------------------	-----------------------------	-------------------	---	-------	-----------------------	-----------------	---------------------------	-------------------	----------------------	--------------------------	---------------------------

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of Partz

				*SKIP = TRUE & *	ILSE S *
•			***	WS==== \$ 30VdS-37000====	+++0RIT 1 SSKIP = FALSE S +
-		**************************************	*CONSTRUCT*LINE * *LINE*NO = LINE*NO + 1 *	S JOHN	
RT2		.1			
DESIGN GIAGRAM OF PARTZ	111 FROM 105	FOR I # F4"START. L. F4"STOP B			

. .. IF A BLANK LINE HAS BEEN . LEFT. INSERT BOX SIDES ..

\*\* SKINE\*NO EQ BOTTON\*LINE - 1 \* \*\* SKIP = FALSE \$

\* SKIP = FALSE \$

\* \*\* SKIP = FALSE \$

\* SKI

\*\*\*\*\*\* \*---\*TRAVERSE"TOP = TRAVERSE"TOP - \* \* 1 \$ \*CUR\*REC = TRAVERSE\*STACK(STRAV\* \*ERSE\*TOP\$) \$ \*IF H\*SON LS D \$ \*\*\*-\*DISPLAY\*STUMP\*REF(L) \$ \* \*EC)\$) \$
\*TRAVERSE\*TOP = TRAVERSE\*TOP - . . . . . \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* \* \*\* IF THERE'S 4 HORIZ STUMP AND A NONSTUMP VSON \*\* \*CUR\*REC = V\*SON \$
\*TRAVERSE\*TOP = TRAVERSE\*TOP \*DO WHILE [V\*PTR(\$ACCESS3(CUR\*R \* \* \*EC)\$) LQ 0 AND CUR\*REC GR 01 \*\* \* \* C S DRAPER LABURATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of partz \*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\* . . . A STUNP .. ---ORIF V-SON GR 0 \$ -----\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* . ......... . 112 FROM 109 \* \*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of partz'init

***** I = 0, 1, TITLE"NO \$ ***		1.97IS-17E - 1 - 37IS-1000000000000000000000000000000000000	I = 0, 1, 001*BUF*SIZE - 1 * ********************************	I = 0, 1, 00I*BUF*SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0, 1, 00I BUF SIZE - 1 * ********************************	I = 0, 1, 001*BUF*SIZE - 1 * ********************************	I = 0, 1, OUT-BUF'SIZE - 1 * ********************************	I = 0, 1, OUT-BUF-SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0, 1, 00I BUF SIZE - 1 * ********************************	I = 0, 1, OUT-BUF-SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************
**************************************		I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT BUF SIZE - 1 *	I = 0, 1, OUT BUF SIZE - 1	I = 0, 1, OUT BUF SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, OUT BUF SIZE - 1	I = 0, 1, OUT BUF SIZE - 1 *	I = 0, 1, OUT BUF SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT BUF SIZE - 1	I = 0, 1, OUT BUF SIZE - 1 *	I = 0, 1, 0UT'BUF'SIZE - 1 *
I = 0, 1, OUT BUF SIZE - 1														
I = 0, 1, 001*80F*SIZE - 1														
I = 0, 1, OUT'BUF'SIZE - 1														
I = 0, 1, OUT'BUF'SIZE - 1					:				***************************************	•••••••••••••••••••••••••••••••••••••••		***************************************	***************************************	
I = 0+ 1+ 0UT'BUF'SIZE - 1 *		00000000000000000000000000000000000000			****									
I = 0, 1, OUT BUF'SIZE - 1		经转换 计对数 计对数 计对数 计对数 计对数 计可数 计可数 计可数 计可数 计可数 计可数 计可数 计可数 计可数 计可												
I = 0, 1, OUT BUF SIZE - 1	THE RESERVENCE OF THE PARTY OF				•						•			
I = 0, 1, 001'80F'SIZE - 1 *	THE PERSON NAMED IN COLUMN TO SECOND	( )HT = (\$18) = TH( )									•			
I = 0, 1, OUT'BUF'SIZE - 1 *	The state of the s				***************************************									
T = 0. 1. Outsalf-SIZE = 1 *	1 7710 100 100 17 10 1													
	I = 0. 1. OUT'BUF'SIZE - 1 *							***************************************						
****	T - A CHITCOLLEGETTE B	1 = 0. 1. 001 BUF SIZE - 1	1 = 0, 1, 00' - 80' - 512E - 1		1 = 371C. 100 - 10 = 1	0				1 - 3715 inp. inp. it in = 1		0 1 - 3716 - 100 - 10 - 10 - 10 - 10 - 10 - 10 -	1 - 3710, 109, 100 of 60 = 1	1 - 3715, 109, 100 of 60 = 1
		I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, 00T'8UF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, 00T'8UF'SIZE - 1 *	I = 0, 1, OUT BUF SIZE - 1
		I = 0, 1, OUT BUF SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, 001°BUF'SIZE - 1 *
		I = 0, 1, OUT BUF SIZE - 1 *	I = 0, 1, OUT BUF SIZE - 1	I = 0, 1, 001*80F*SIZE - 1 *	I = 0, 1, OUT BUF SIZE - 1	I = 0, 1, OUT BUF SIZE - 1	I = 0, 1, 001*80F*SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, OUT BUF SIZE - 1	I = 0, 1, OUT BUF SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, OUT'BUF'SIZE - 1 *
		I = 0, 1, 00T*8UF*SIZE - 1 *	I = 0, 1, 001°80F°SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, 00T'BUF'SIZE - 1	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, 00T'8UF'SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1 *
		I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT BUF SIZE - 1 *	I = 0, 1, OUT BUF SIZE - 1	I = 0, 1, OUT BUF SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, OUT BUF SIZE - 1	I = 0, 1, OUT BUF SIZE - 1 *	I = 0, 1, OUT BUF SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, OUT BUF SIZE - 1	I = 0, 1, OUT BUF SIZE - 1 *	I = 0, 1, 0UT'BUF'SIZE - 1 *
		I = 0, 1, OUT'8UF'SIZE - 1	I = 0, 1, 00T'8UF'SIZE - 1 *	I = 0, 1, OUT BUF'SIZE - 1	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, OUT BUF'SIZE - 1	I = 0, 1, OUT BUF'SIZE - 1 *	I = 0, 1, 0u1*8uf*SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, OUT BUF SIZE - 1 *	I = 0, 1, 0UT'BUF'SIZE - 1	I = 0, 1, OUT BUF'SIZE - 1	I = 0, 1, 0UT'BUF'SIZE - 1	I = 0, 1, 00T'BUF'SIZE - 1
		I = 0, 1, 0UT'8UF'SIZE - 1 *	I = 0, 1, OUT*BUF*SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, OUT BUF SIZE - 1	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, 0UT*BUF*SIZE - 1	I = 0, 1, OUT BUF'SIZE - 1	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, 001*80F*SIZE - 1	I = 0, 1, OUT BUF SIZE - 1 *	I = 0, 1, OUT'BUF'SIZE - 1	I = 0, 1, 0UT'BUF'SIZE - 1 *
		I = 0, 1, OUI'BUF'SIZE - 1 P ercessesses	I = 0, 1, 001°80F°SIZE - 1 * ********************************	I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	I = 0, 1, OUT-BUF-SIZE - 1	I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	I = 0, 1, 0UT'BUF'SIZE - 1 * ********************************	I = 0, 1, 0UT'BUF'SIZE - 1 * ********************************	I = 0+ 1+ 00T°BUF°SIZE - 1 + ++++++++++++++++++++++++++++++++	I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	I = 0, 1, 0UT'BUF'SIZE - 1 * ********************************	I = 0, 1, OUT'BUF'SIZE - 1 **********************************
		I = 0+ 1+ Out-Buf-Size - 1 + extenses	I = 0, 1, our surrecesses as seconsesses I = 0, 1, our surrecesses as seconsesses as seconses as	I = 0, 1, 00T*BUF*SIZE - 1 * ********************************	I = 0. 1. OUT BUF SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0+ 1+ 00T*BUF*SIZE - 1 * ********************************	I = 0+ 1+ OUT*BUF*SIZE - 1 * ********************************	I = 0, 1, OUT'BUF'SIZE - 1 * -+000T'LINE(\$1\$)	I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	I = 0, 1, 0UT'BUF'SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0+ 1+ OUT*BUF*SIZE - 1 * ********************************	I = 0, 1, 0UT°BUF°SIZE - 1 * ********************************
	**********************	I = 0, 1, 001°BUF'SIZE - 1 * ********************************	######################################	I = 0, 1, Out Buf Size - 1 + cessessessessessessessessessessessessess	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0, 1, Our Buf Size - 1 + ++++++++++++++++++++++++++++++++	I = 0, 1, Out Buf Size = 1 + ++++++++++++++++++++++++++++++++	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	I = 0, 1, Out Buf Size - 1 + cerestresses	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0, 1, 00T*BUF*SIZE - 1 * ********************************
		I = 0, 1, 0UT'BUF'SIZE - 1 * ********************************	######################################	I = 0, 1, 0UT°BUF°SIZE - 1 * ********************************	I = 0, 1, 001*80F*SIZE - 1 * ********************************	I = 0, 1, OUT°BUF°SIZE - 1 **********************************	I = 0, 1, 0UT*BUF*SIZE - 1 * ********************************	I = 0, 1, 0UT*BUF*SIZE - 1 * ********************************	I = 0, 1, OUT'BUF'SIZE - 1 **********************************	I = 0, 1, 0UT'BUF'SIZE - 1 * ********************************	######################################	I = 0, 1, OUT BUF SIZE - 1 **********************************	I = 0, 1, 0UT'BUF'SIZE - 1 * ********************************	I = 0, 1, 0UT'BUF'SIZE - 1 * ********************************
	****	I = 0 + 1 + 0.01 + 0.01 = 0.01 = 0.01	I = 0. 1. OUT BUF SIZE - 1 * ********************************	I = 0, 1, 0UT'BUF'SIZE - 1 * ********************************	I = 0, 1, OUT-BUF-SIZE - 1 * ********************************	I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	I = 0, 1, 00T BUF SIZE - 1 * ********************************	I = 0+ 1+ OUT'BUF'SIZE - 1 * ********************************	I = 0+ 1+ 00T*BUF*SIZE - 1 *+00U**LNE(818)	I = 0+ 1+ OUT*BUF*SIZE = 1 * ********************************	I = 0, 1, 0UT°BUF°SIZE - 1 * ********************************	I = 0, 1, OUT'BUF'SIZE - 1 **********************************	I = 0+ 1+ 001*BUF*SIZE - 1 *+001*LINE(818)	I = 0, 1, OUT'BUF'SIZE - 1 * ********************************
S TOTAL T I US IN INITE NO S TOTAL	***************************************	I = 0, 1, OUT'BUF'SIZE - 1 *	I = 0, 1, 001*BUF*SIZE - 1 * ********************************	I = 0, 1, 0UI*BUF*SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0, 1, 00I*BUF*SIZE - 1 * ********************************	I = 0, 1, 0UI*BUF*SIZE - 1 * ********************************	I = 0, 1, OUT-BUF'SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0, 1, 0UT'BUF'SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************
S STATE OF T S D. 4. TITIE'NO C CO.		I = 0, 1, OUT-BUF-SIZE - 1 *	I = 0, 1, 0UT'8UF'SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1	I = 0, 1, 0UT'BUF'SIZE - 1 * ********************************	I = 0, 1, 0UT'BUF'SIZE - 1 **********************************	I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	I = 0, 1, OUT'BUF'SIZE - 1 **********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0, 1, 0UT'BUF'SIZE - 1 * ********************************	I = 0, 1, OUT-BUF-SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************	I = 0, 1, OUT BUF SIZE - 1 * ********************************
	everence I = 0, 1, IILE NO S	ITLE SM 8FOR I = 0, 1, TITLE NO 8 received and in the state of the state o	ITLE SN 8FOR I = 0, 1, IILLE NO 8 spectate to the control of the control	ITLE SN & *****FOR I = 0, 1, TILLE NO S ***  FREE CONTROL STATE   ***********************************	ITLE SN 8FOR I = 0, 1, TILLE NO 8  I = 0, 1, OUT BUF SIZE - 1	ITLE SN & **********************************	ITLE SN 8FOR I = 0, 1, TILLE NO 8  FERRESSEE - 1 - 0.1, OUT BUF SIZE - 1 - 0.1, TREE (SIS  FERRESSEE - 1 - 0.1, TREE  FERRE	ITLE SN 8FOR I = 0, 1, TILLE NO 8  RESERVANCE CONTROL SIZE - 1 * ********************************	ITLE SM & *****FOR I = 0, 1, TITLE NO & ****  I = 0, 1, OUT BUF SIZE - 1 * ********  ***********************	TILE'SN & *****FOR I = 0, 1, TILE'NO & ***  PREFERENCE   F   F   F   F   F   F   F   F   F	TILE'SN & *****FOR I = 0, 1, TILE'NO & ***  PERSONAL PROPERTY OF STREET	ITLE SN & *****FOR I = 0, 1, IILLE NO S ***  FOR THE STATE OF THE STAT	ITLE SM SFOR I = 0, 1, TILLE NO S  PERSONNEL STREET STRE	ITLE SN 8FOR I = 0, 1, IIILE NO 8
	**************************************	ITLE*SN \$ **********************************	ITLE'SM 8 *****FOR I = 0, 1, IIILE'NO 8 ***  *******************************	ITLE'SM 8FOR I = 0, 1, TITLE'NO 8  THE STATE OF THE STATE OF THE STATE OF THE STATE  THE STATE OF TH	ITLE SN 8FOR I = 0, 1, TITLE NO 8  STATE OF 1, OUTBUF SIZE - 1  T = 0, 1, OUTBUF SIZE - 1  STATE OF 1, OUTBUF SIZE - 1	ITLE'SN SFOR I = 0, 1, TITLE'NO S stressessessessessessessessessessessessess	ITLE'SM 8FOR I = 0, 1, TITLE'NO 8  THE STATE OF THE STATE I  THE STATE I	ITLE'SM & "FOR I = 0, 1, TITLE'NO & " stressessessessessessessessessessessessess	ITLE'SN & GFOR I = 0, 1, TITLE'NO & F stressessessessessessessessessessessessess	ITLE SN SFOR I = 0, 1, TITLE NO S  ********************************	ITLE SN 8FOR I = 0, 1, IIILE NO 8  I = 0, 1, OUT BUF SIZE - 1	ITLE SN 8FOR I = 0, 1, TITLE NO 8 stressessessessessessessessessessessessess	ITLE SN SFOR I = 0, 1, TITLE NO S statements of the statement of	ITLE SN SFOR I = 0, 1, TITLE NO S  STATE OF THE STATE OF
	**************************************	ITLE'SM &FOR I = 0, 1, TITLE'NO & eresessesses eresessessessessessesses I = 0, 1, 001'80F'SIZE - 1 *	ITLE SM \$ ****FOR I = 0, 1, TITLE NO \$ ***  *******************************	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$ stressessessessessessessessessessessessess	ITLE SN \$FOR I = 0, 1, TITLE NO \$  FREE STATE	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$ stressessessessessessessessessessessessess	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$ stressessessessessessessessessessessessess	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$ secretary secretary I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	ITLE'SM &FOR I = 0, 1, TITLE'NO &  FREE STATE	ITLE'SM &FOR I = 0, 1, TITLE'NO & eresesses eresessesses  I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	IILE'SM \$ *****FOR I = 0, 1, TITLE'NO \$ ****  ***************************  I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	ITLE SN SFOR I = 0, 1, TITLE NO S  STATE OF THE	ITLE'SM &FOR I = 0, 1, TITLE'NO &  server and a	IILE'SN \$ ****FOR I = 0, 1, TITLE'NO \$ ****  I = 0, 1, OUT'BUF'SIZE - 1 * *********  **********************
	**************************************	ITLE SM & ***** FOR I = 0, 1, TITLE NO & ****  ******************************	ITLE SH & **********************************	ITLE SM &FOR I = 0, 1, TITLE NO &  ELECTRICAL STATE	ITLE SH & **********************************	ITLE SM 8FOR I = 0, 1, TITLE NO 8  ELECTRICAL STATES IN THE STATES I = 0, 1, OUT BUF SIZE - 1 * ********************************	ITLE SM &FOR I = 0, 1, TITLE NO &  ELECTRICAL STATE	ITLE SM SFOR I = 0, 1, TITLE NO S  ELECTRICAL STATE	ITLE SM SFOR I = 0, 1, TITLE NO S  I = 0, 1, OUT BUF SIZE - 1 * ********************************	ITLE SM & ***** FOR I = 0, 1, TITLE NO & ****  ******************************	ITLE SH & **********************************	ITLE SH & ***** FOR I = 0, 1, IIILE NO & ****  I = 0, 1, OUT BUF SIZE - 1 *******  **************************	ITLE SM SFOR I = 0, 1, TITLE NO S  I = 0, 1, OUT BUF SIZE - 1 * ********************************	ITLE SH & **********************************
*******************	ecccicacecccccccccccccccccccccccccccccc	ITLE SN SFOR I = 0, 1, TITLE NO S statements of the statement of t	ITLE'SM & ***********************************	ITLE'SM & *****FOR I = 0, 1, TITLE'NO & ****  ******************************	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  FREE SM \$FOR I = 0, 1, TITLE'NO \$  I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	ITLE'SM & *****FOR I = 0, 1, TITLE'NO & ****  Free control of the	ITLE'SM & *****FOR I = 0, 1, TITLE'NO & ****  ******************************	ITLE'SN S *****FOR I = 0, 1, TITLE'NO S ***********************************	ITLESM S GFOR I = 0, 1, TITLE NO S GFOR THE FORM S GFOR THE FORM S GFORT T	ITLESM & ***********************************	ITLESM \$FOR I = 0, 1, TILLE'NO \$  FREE STATES OF THE STATES OF	ITLESM SFOR I = 0, 1, TITLE'NO S  PERSONNEL - 1	ITLESM S GFOR I = D, 1, TITLE NO S GFOR	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  I = 0, 1, OUT'BUF'SIZE - 1 * ********************************
:	**************************************	ITLE'SM 8 *****FOR I = 0, 1, IIILE'NO 8 ***  *******************************	ITLESM SFOR I = 0, 1, TILLE'NO S statement of the control	ITLESH &FOR I = 0, 1, TITLE NO S * state of the control of t	ITLESN SFOR I = 0, 1, TITLE NO S  THE STATE OF TH	ITLESN SFOR I = 0, 1, TITLE NO S state of the control of t	ITLESH &FOR I = 0, 1, IIILE'NO & second and in the control of the control	ITLE'SH & "FOR I = 0, 1, IIILE'NO & " stressesses	ITLE'SM & ****FOR I = 0, 1, IIILE'NO & ****  I = 0, 1, OUI'BUF'SIZE - 1 ****  *****************************	ITLE SN 8FOR I = 0, 1, TITLE NO 8 PERSON SFOR I = 0, 1, TITLE NO 8  I = 0, 1, OUT BUF SIZE - 1	ITLESN SFOR I = 0, 1, TITLE NO S  THE STATE OF T	ITLESSW SFOR I = 0, 1, TITLE NO S statement of the statement of th	ITLE'SM & "FOR I = 0, 1, IIILE'NO & " statement of the contract of the co	ITLE SN 8FOR I = 0, 1, TITLE NO 8 ITLE SN 1, OUTBUF SIZE - 1 * ********************************
******************	**************************************	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$ errererer  THE SM SFOR I = 0, 1, TITLE'NO \$ ERRERERER  THE SM SFOR I = 0, 1, TITLE'NO \$ ERRERERERERERERERERERERERERERERERERE	ITLE'SM & GFOR I = 0, 1, IIILE'NO & GFOR I = 0, IIILE'NO &	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  ETTERSON \$FOR I = 0, 1, TITLE'NO \$  ETTERSON \$FOR I = 0, 1, TITLE'NO \$  ETTERSON \$FOR I = 0, 1, TITLE'NO \$  I = 0, 1, OUT BUF'SIZE - 1 * ********************************	ITLE SN 8FOR I = 0, 1, TITLE NO 8  ETTE SN 8FOR I = 0, 1, TITLE NO 8  ETTE SN 8FOR I = 0, 1, TITLE NO 8  ETTE SN 1  ETT	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  ETTE: SM \$FOR I = 0, 1, TITLE'NO \$  ETTE: SM \$FOR I = 0, 1, TITLE'NO \$  ETTE: SM \$FOR I = 0, 1, TITLE'NO \$  I = 0, 1, OUT BUF'SIZE - 1 * ********************************	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$ stressessessessessessessessessessessessess	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$ secretaring a s	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  Exercises to the control of the contr	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  FREE STATE	ITLE SM SFOR I = 0, 1, TITLE NO S  Extrements continue to the continu	ITLE'SM &FOR I = 0, 1, TITLE'NO S  ETTERSON SFOR I =	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  Exercise to the contract of the co	ITLE SM \$FOR I = 0, 1, TITLE NO \$  ETTE SM \$FOR I = 0, 1, TITLE NO \$  ETTE SM \$FOR I = 0, 1, TITLE NO \$  I = 0, 1, OUT BUF SIZE - 1 * ********************************
	**************************************	ITLE SW 8FOR I = 0, 1, TITLE NO 8 stressess control of the control of th	ITLE'SM S "FOR I = 0, 1, IIILE'NO S "  SECRETARIST STATE STAT	IILE SM 8FOR I = 0, 1, TITLE NO 8  ETHERSON SFOR I = 0, TITLE NO 8  ETHERSON SFOR I = 0, TITLE NO 8  ETHERS	ITLE'SH & *****FOR I = 0, 1, IIILE'NO & ***********************************	ITLE SM 8FOR I = 0, 1, TITLE NO 8  ETHERS SM 8FOR I = 0, 1, TITLE NO 8  ETHERS SM 8FOR I = 0, 1, TITLE NO 8  ETHERS SM 8FOR I = 0, 1, TITLE NO 8  ETHERS SM 8FOR I = 0, 1, TITLE NO 8  ETHERS SM 8FOR I = 0, 1, TITLE NO 8  ETHERS SM 8FOR I = 0, 1, TITLE NO 8  ETHERS SM 8FOR I = 0, 1, TITLE NO 8  ETHERS SM 8FOR I = 0, 1, TITLE NO 8  ETHERS SM 8FOR I = 0, 1, TITLE NO 8  ETHERS SM 8FOR I = 0, 1, TITLE NO 8  ETHERS SM 8FOR I = 0, 1, TITLE NO 8  ETHERS SM 8FOR I = 0, 1, TITLE NO 8  ETHERS SM 8FOR I = 0, TITL	IILE SM 8FOR I = 0, 1, TITLE NO 8  Exercises to the contract of the c	ITLE SW 8FOR I = 0, 1, TITLE NO 8  Exercises to a control of the control	ITLE SW 8FOR I = 0, 1, TITLE NO 8  ETT SW 1, OUT BUF SIZE - 1 * * * * * * * * * * * * * * * * * *	ITLE SW 8FOR I = 0, 1, TITLE NO 8  SECOND SW 1	ITLE'SH & *****FOR I = 0, 1, IIILE'NO & ***********************************	ITLE SM &FOR I = 0, 1, TITLE NO &  ETTE SM &FOR I =	ITLE SW 8FOR I = 0, 1, TITLE NO 8  ETT ST S	ITLESH & ***********************************
*****	servers I = 0, 1, TITE NO S	ITLESH & ***********************************	ITLE'SM & Garage Control of the Cont	ITLE'SM & GFOR I = 0, 1, TITLE'NO & G RECENSES GFOR I = 0, 1, TI	ITLE'SM & GFOR I = 0, 1, TITLE'NO & G  THE SM & GFOR I = 0, 1, TITLE'NO & G  THE SM & GFOR I = 0, 1, TITLE'NO & G  THE SM & GFOR I = GFOR I I G  THE SM & GFOR I = GFOR I I I G  THE SM & GFOR I = GFOR I I G  THE SM & GFOR I = GFOR I I G  THE SM & GFOR I = GFOR I I G  THE SM & GFOR I = GFOR I I G  THE SM & GFOR I = GFOR I I G  THE SM & GFOR I = GFOR I I G  THE SM & GFOR I = GFOR I I G  THE SM & GFOR I = GFOR I I G  THE SM & GFOR I = GFOR I I G  THE SM & GFO	ITLE'SM & GFOR I = 0, 1, TITLE'NO & G ESTERNATE CONTRIBUTION OF THE STATE OF T	ITLESH & ***********************************	ITLE SM & **** FOR I = 0, 1, IIILE NO & ****  I = 0, 1, OUT BUF SIZE - 1 ****  *****************************	ITLE'SM & "FOR I = 0, 1, IIILE'NO & "  SECOND IN THE	ITLE SM SFOR I = 0, 1, TITLE NO S  THE SM SFOR I = 0, 1, TITLE NO S  I = 0, 1, OUT BUF SIZE - 1 * ********************************	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  IT = 0, 1, OUT'BUF'SIZE - 1 * *OUT'LINE(SIS)	ITLE'SM & Garage Control of the Cont	ITLE'SM & "FOR I = 0, 1, IIILE'NO & " ITLE'SM & "FOR I = 0, 1, IIILE'NO & " I = 0, 1, OUT'BUF'SIZE - 1 " "	ITLE'SM \$ "FOR I = 0, 1, TITLE'NO \$ " ITLE'SM \$ "FOR I = 0, 1, OUT BUF'SIZE - 1 "FOUT'LINE(SIS
***************************************	eresessessessessessessessessessessessess	ITLE'SH & "FOR I = 0, 1, IITLE'NO & " stressesses exceptions of the stresses of the stre	ITLE'SN S *****FOR I = 0, 1, TITLE'NO S ****  ******************************	ITLE'SN S ****FOR I = 0, 1, TITLE'NO S ****  I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	ITLE'SN S*FOR I = 0, 1, TITLE'NO S *  TENTE OF 1, OUT-BUF'SIZE - 1 * ********************************	ITLE'SN SFOR I = 0, 1, TITLE'NO S "  PERSONNEL CONTRICTOR OF THE PERSONNEL CON	ITLE SN 8 ****FOR I = 0, 1, TITLE NO 8 ****  I = 0, 1, OUT BUF SIZE - 1 * ********  ***********************	ITLE SM &FOR I = 0, 1, TITLE NO S *  THE SM STATE STA	ITLE'SM & GARDEN STREET	ITLE'SM & ***********************************	ITLE'SM & ***********************************	ITLE SN SFOR I = 0, 1, TITLE NO S  PERSONNEL CONTRIBUTION S STATEMENT OF S	ITLE SM S GFOR I = 0, 1, TITLE NO S  FREE FOR S GFOR I = 0, 1, TITLE NO	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  STATE OF THE PROPERTY OF THE PROPE
	**************************************	ITLE'SM \$ "FOR I = 0, 1, TITLE'NO \$ " state of the contraction of	ITLE'SN &FOR I = 0, 1, TILLE'NO & state of the contract of the contrac	ITLESH & ***********************************	ITLE'SW SFOR I = 0, 1, TITLE'NO S I = 0, 1, OUT'BUF'SIZE - 1	ITLESH & ***********************************	ITLESM & ***********************************	ITLE'SN 8FOR I = 0, 1, TITLE'NO 8  I = 0, 1, OUT BUF SIZE - 1OUT LINE(SIZE - 1OUT LINE(SIZE - 1OUT LINE(SIZE - 1OUT LINE(SIZE - 1	ITLE SW SFOR I = 0, 1, TITLE NO S  I = 0, 1, OUT BUF SIZE - 1	ITLE SN 8 ****FOR I = 0, 1, TITLE NO 8 ***  I = 0, 1, OUT BUF SIZE - 1 **********************************	ITLE'SN 8FOR I = 0, 1, TITLE'NO 8  I = 0, 1, OUT'BUF'SIZE - 1	ITLESH &FOR I = 0, 1, TITLE NO 8 state of the control of t	ITLE SM 8FOR I = 0, 1, TITLE NO 8  THE SM 1, THE SM 1, TITLE NO 8  I = 0, 1, OUT BUF SIZE - 1OUT LINE (SIS	ITLE SN 8FOR I = 0, 1, TITLE NO 8 IT = 0, 1, OUT BUF SIZE - 1OUT LINE(SIS
	**************************************	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$ FORTHWELL STATES OF THE STATE	ITLE SN 8FOR I = 0, 1, TITLE NO 8  STATE OF 1, OUT BUF SIZE - 1	ITLE'SM SFOR I = 0, 1, TITLE'NO S  I = 0, 1, OUT BUF'SIZE - 1 * ********************************	ITLE SN SFOR I = 0, 1, TITLE NO S  I = 0, 1, OUT BUF SIZE - 1	ITLE'SM 8FOR I = 0, 1, TITLE'NO 8  ITLE'SM 8FOR I = 0, 1, TITLE'NO 8  I = 0, 1, OUT BUF'SIZE - 1 * ********************************	ITLE'SM \$ "FOR I = 0, 1, TITLE'NO \$ "  ITLE'SM \$ "FOR I = 0, 1, TITLE'NO \$ "  I = 0, 1, OUT BUF'SIZE - 1 " "FOUT LINE(RIS	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  Extraction of the contraction of the contract	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  FREE STATE	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  FREE FOR THE FOR T	ITLE SW 8FOR I = 0, 1, TITLE NO 8  I = 0, 1, OUT BUF SIZE - 1	ITLE SN SFOR I = 0, 1, TITLE NO S  E TO 1, TO UT BUF SIZE - 1 * ********************************	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  Extraction of the contraction of the contract	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$ ITLE'SM \$FOR I = 0, 1, TITLE'NO \$ I = 0, 1, OUT'BUF'SIZE - 1 * ********************************
	**************************************	ITLE SW 8FOR I = 0, 1, TITLE NO 8 ETERORESE CONTRACTOR OF THE PROPERTY OF THE PROPERT	ITLE'SM & ***********************************	ITLE SN 8FOR I = 0, 1, TITLE NO 8  ETTE SN 8FOR I =	ITLE'SM S "FOR I = 0, 1, TITLE'NO S " I = 0, 1, OUT'BUF'SIZE - 1 " "OUT'LINE(SIS	ITLE SN 8FOR I = 0, 1, TITLE NO 8  ETTE SN 8FOR I =	ITLE SN 8FOR I = 0, 1, TITLE NO 8  ETTE SN 8FOR I =	ITLE SW 8FOR I = 0, 1, TITLE NO 8  E D, 1, OUT-BUF'SIZE - 1 **********************************	ITLE SW 8FOR I = 0, 1, TITLE NO 8  I = 0, 1, OUT BUF SIZE - 1 * ********************************	ITLE SW 8FOR I = 0, 1, TITLE NO 8  THE SW 8FOR I = 0,	ITLE'SH & "FOR I = 0, 1, IIILE'NO & "  THE SH & "FOR I = 0, 1, IIILE'NO & "  THE SH & "FOR I = 0, 1, IIILE'NO & "  THE SH & "FOR I = 0, 1, IIILE'NO & "FOR I = 0, IIILE'NO & "FOR I =	ITLE SN 8FOR I = 0, 1, TITLE NO 8  ETTE STATE	ITLE SW &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &FOR I = 0, 1, TITLE NO &  ETHERSON &	ITLESH & ***********************************
	TARREST TO TARREST TAR	ITLESH & ***********************************	ITLE'SM & ***********************************	ITLE SW SFOR I = 0, 1, TITLE NO S  ETTE SW SFOR I = 0, TITLE NO S  ETTE SW SFOR I = 0, TITLE NO S	ITLE SM S GFOR I = 0, 1, TITLE NO S G  I = 0, 1, OUT BUF SIZE - 1 GOUT LINE (SIS	ITLE'SM & ***********************************	ITLE SW SFOR I = 0, 1, TITLE NO S  ETTE SW SFOR I = 0, TITLE NO S  ETTE SW SFOR I = 0, TITLE NO S	ITLESS SFOR I = 0, 1, TITLE'NO S  I = 0, 1, OUT BUF'SIZE - 1	ITLE'SM S **********************************	ITLE'SM SFOR I = 0, 1, TITLE'NO S I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$ I = 0, 1, OUT'BUF'SIZE - 1 * *OUT'LINE(SIS	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$ FORTHWENT PROPERTY AND THE PROP	ITLE'SM & *****FOR I = 0, 1, IIILE'NO & ***********************************	ITLE'SM \$ "FOR I = 0, 1, TITLE'NO \$ " I = 0, 1, OUT BUF'SIZE - 1 "
1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	* + 1 S * * * 1 S * * * * * * * * * * * * * * * * * * *	ITLE SM 8FOR I = 0, 1, IIILE NO 8 Exercises to the second secon	ITLE SN 8FOR I = 0, 1, TITLE NO 8  ETTE SN 8FOR I = 0, 1	ITLE SW SFOR I = 0, 1, TITLE NO S  THE SW	ITLE SN 8FOR I = 0, 1, TITLE NO 8  I = 0, 1, OUT BUF SIZE - 1	ITLE SW SFOR I = 0, 1, TITLE NO S ITLE SW SFOR I = 0, 1, TITLE NO S IT = 0, 1, OUT BUF SIZE - 1	ITLE'SW SFOR I = 0, 1, TITLE'NO S STATEST STAT	ITLESH & ****FOR I = 0, 1, TITLE NO & ****  ITLESH & *****  I = 0, 1, OUT BUF SIZE - 1 *****  ******  ******  ******  *****  ****	ITLESSM &FOR I = 0, 1, TITLE'NO & ITLESSM &FOR I = 0, 1, TITLE'NO & I = 0, 1, OUT'BUF'SIZE - 1 * ********************************	ITLE'SM & ***********************************	ITLE SN SFOR I = 0, 1, TITLE NO S  ITLE SN SFOR I = 0, 1, TITLE NO S  I = 0, 1, OUT BUF SIZE - 1	ITLE SW SFOR I = 0, 1, TITLE NO S ITLE SW SFOR I = 0, 1, TITLE NO S IT = 0, 1, OUT BUF SIZE - 1	ITLE SW 8 "FOR I = 0, 1, IIILE NO 8 "  II = 0, 1, OUT BUF SIZE - 1 " "********************************	ITLE'SM &FOR I = 0, 1, TITLE'NO &  TOUT BUF'SIZE - 1 * ********************************
S	* 1 \$	ITLE'SM \$FOR I = 0, 1, IITLE'NO \$ second and a s	ITLE'SH & ****FOR I = 0, 1, ITLE'NO & ****  ******************************	ITLE SW SFOR I = 0, 1, TITLE NO S I = 0, 1, OUT BUF SIZE - 1FOUT LINE(SIS	ITLE'SW F. FOR I = 0, 1, TITLE'NO S F. II = 0, 1, OUT'BUF'SIZE - 1 F.	ITLE SW SFOR I = 0, 1, TITLE NO S I = 0, 1, OUT BUF SIZE - 1 * ********************************	ITLE'SM SFOR I = 0, 1, TITLE'NO S I = 0, 1, OUT BUF'SIZE - 1	ITLE*SM \$ *FOR I = 0, 1, TITLE*NO \$ *  I = 0, 1, OUT*BUF*SIZE - 1 **********************************	ITLE SM 8FOR I = 0, 1, ITLE NO 8  ETTE SM 8FOR I = 0, ITLE NO 8  ETTE SM	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$ FORTHWELL I = 0, 1, OUT BUF'SIZE - 1 FOUT LINE(SIZE - 1 FOUT	ITLESH &FOR I = 0, 1, TITLE NO &  I = 0, 1, OUT-BUF-SIZE - 1	ITLE SW SFOR I = 0, 1, TITLE NO S  I = 0, 1, OUT BUF SIZE - 1 * ********************************	ITLE SM 8FOR I = 0, 1, ITLE NO 8  Extraction of the contraction of the contracti	ITLE SW SFOR I = 0, 1, TITLE NO S  I = 0, 1, OUT BUF SIZE - 1
	TENDER I BOY LATER THE TENDER OF THE TENDER	ITLE SN 8FOR I = 0, 1, TITLE NO 8 ETTERSON 8	ITLE SM 8FOR I = 0, 1, IIILE NO 8  Exercise to 1, OUT BUF SIZE - 1 * ********************************	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  ETTERS STATE OF THE STATE	ITLE SN 8FOR I = 0, 1, TITLE NO 8  E TO 1, TO UT BUF SIZE - 1	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  ETTERS B	ITLE SM 8FOR I = 0, 1, TITLE NO 8  ETTERS STATE STA	ITLE SN 8FOR I = 0, 1, TITLE NO 8  ETTER SN 8FOR I = 0, TITLE NO 8  ETTER S	ITEESN SFOR I = 0, 1, TITEENO S  I = 0, 1, OUT-BUF-SIZE - 1 **********************************	ITEE'SW SFOR I = 0, 1, TITEE'NO S  THE STATE OF	ITLESSW & **********************************	ITLE SM 8FOR I = 0, 1, TITLE NO 8  ETTE STATE	ITLE SW 8FOR I = 0, 1, TITLE NO 8  I = 0, 1, OUT-BUF-SIZE - 1 **********************************	ITLESH & ***********************************
* LENGIM(HEADER(SNAME'INDEX\$))*  * * 1	* LENGTH(HEADER(SNAME*INDEXS))*  * 1	TILE SH & FFOR I = 0, 1, IILE NO & FFOR I = 0, 1, IIILE NO & FFOR I = 0, IIILE NO & FFOR I =	ITLE'SM & ***********************************	ILLE'SW &FOR I = 0, 1, TITLE'NO &  ETTER SW &FOR I =	ITLE'SM & "FOR I = 0, 1, TITLE'NO & "  ETTESM & "FOR I = 0, 1, TITLE'NO & "  ETTESM & "FOR I = 0, 1, TITLE'NO & "  ETTESM & "FOR I = 0, 1, TITLE'NO & "  ETTESM & "FOR I = 0, 1, TITLE'NO & "FO	LENGIN(HEADER(SNAME INDEXS))  * 1	ILLE'SW &FOR I = 0, 1, TITLE'NO &  ETTESW &FOR I = 0,	ILESS SFOR I = 0, 1, IILE'NO S  TREES S SFOR I = 0, 1, IILE'NO S S  TREES S S S S S S S S S S S S S S S S S S	ILLE'SM & "FOR I = 0, 1, IIILE'NO & "  ILLE'SM & "FOR I = 0, 1, IIILE'NO & "  ILLE'SM & "FOR I = 0, 1, IIILE'NO & "  ILLE'SM & "FOR I = 0, 1, IIILE'NO & "  ILLE'SM & "FOR I = 0, 1, IIILE'NO & "  ILLE'SM & "FOR I = 0, 1, IIILE'NO & "  ILLE'SM & "FOR I = 0, 1, IIILE'NO & "  ILLE'SM & "FOR I = 0, 1, IIILE'NO & "  ILLE'SM & "FOR I = 0, 1, IIILE'NO & "  ILLE'SM & "FOR I = 0, 1, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'NO & "  ILLE'SM & "FOR I = 0, IIILE'SM & "  ILLE'SM	TILE'SM & "FOR I = 0, 1, TITLE'NO & "  ITLE'SM & "FOR I = 0, 1, TITLE'NO & "  ITLE'SM & "FOR I = 0, 1, TITLE'NO & "  I = 0, 1, OUT'BUF'SIZE - 1 " "	ITLE'SM \$ "FOR I = 0, 1, TITLE'NO \$ "  ITLE'SM \$ "FOR I = 0, 1, TI	ITLE'SM & ***FOR I = 0, 1, TITLE'NO & **-  ** I = 0, 1, OUT'BUF'SIZE - 1 **********************************	ILLE'SH & "FOR I = 0, 1, IILE'NO & "  EXECUTE OF THE PROOF OF THE	ITLE'SM \$ "FOR I = 0, 1, TITLE'NO \$ " ITLE'SM \$ "FOR I = 0, 1, TITLE'NO \$ " ITLE'SM \$ "FOR I = 0, 1, TITLE'NO \$ " ITTLE'SM \$ "FOR I = 0, 1, TITLE'NO
* LENGTH(HEADER(SNAME*INDEXS))*  * + 1 S  **********************************	E LENGTH(HEADER(SNAME"INDEXS)) # + 1 & strate to the strat	ELENGTH (HEADER(SNAME INDEXS))  STATES OF THE STATES OF TH	LENGTH(HEADER(SNAME INDEXS))  + 1	LENGTH(HEADER(SNAME'INDEXS))  + 1	LENGTH(HEADER(SNAME INDEXS))  + 1	LENGTH(HEADER(SNAME INDEXS))  + 1	LENGTH(HEADER(SNAME'INDEXS))  + 1	ILLE'SM & "FOR I = 0, 1, TITLE'NO & "  ILLE'SM & "FOR I =	ELENGTH (HEADER(SNAME*INDEXS))  + 1	ELENGTH (HEADER(SNAME"INDEXS))  + 1	LENGTH(HEADER(SNAME*INDEXS))  + 1	LENGTH(HEADER(SNAME INDEXS))  + 1	ELENGTH (HEADER(SNAME"INDEXS))  * 1	ILLESM & ***FOR I = 0, 1, TITLE NO S **-  *** I
LENGTH(HEADER(SNAHE'INDEXS))+ + 1 S + 2 S + 3 S + 4 S	LENGTH (HEADER(SNAME'INDEXS)) + 1 S	ITLE'SM \$FOR I = 0, 1, IITLE'NO \$  ELECTH(HEADER(SNAME'INDEXS))  ITLE'SM \$FOR I = 0, 1, IITLE'NO \$  ELECTHORY  ELE	LENGTH(HEADER(SNAHE'INDEXS)) = 1	ILLE'SM 8 "FOR I = 0, 1, TILLE'NO 8 "  Exercise to the control of the contro	LENGTH(HEADER(SNAHE'INDEXS)) = 1	ILLE'SH & "FOR I = 0, 1, TILLE'NO & "  """  """  """  """  """  """  ""	ILLE'SM 8FOR I = 0, 1, TILLE'NO 8  TELE'SM 8FOR I = 0, 1, TILLE'NO 8  THE TOTAL STATES AND STA	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  I = 0, 1, OUT BUF'SIZE - 1 *OUT LINE(SIZE)	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  ITLE'SM \$FOR I =	LENGTH(HEADER(SNAHE'INDEXS)) • 1 S •	LENGTH(HEADER(SNAHE'INDEXS)) = 1	ILLE'SH & "FOR I = 0, 1, TILLE'NO & "  """  """  """  ""  """  ""  ""  ""	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$  ITLE'SM \$FOR I =	LENGTH(HEADER(SNAHE'INDEXS)) + 1 \$  1 1
TENGIN (HEADER(SNAME'INDEXS))  + 1 S	* LENGTH (HEADER(SNAME'INDEXS))  * + 1	THE SHIP STATE TARGEN FOR STATE STAT	ILLE'SM \$ "FOR I = 0, 1, IILLE'NO \$ "  TELE'SM \$ "FOR I = 0, 1, IILL	ITLE SW SFOR I = 0, 1, IIILE NO S  TENERGRAPHE INDEXS)  TENERGRAPHE INDEXS)  TO I OUT BUF SIZE - I THE TENERGRAPHE  TO I OUT B	ILLE'SM \$ "FOR I = 0, 1, TILLE'NO \$ "  ILLE'SM \$ "FOR I = 0, 1, TILLE'NO \$ "	ITLE SW SFOR I = 0, 1, TITLE NO S  TOTAL SW S	ITLE'SM \$ "FOR I = 0, 1, IIILE'NO \$ "  ITLE'SM \$ "FOR I = 0, IIILE'NO \$ "  ITLE'SM \$ "FOR I = 0, IIILE'NO \$ "  ITLE'SM \$ "FOR I = 0, IIILE'NO \$ "  ITLE'SM \$ "FOR I = 0, IIILE'	ITLE SM &FOR I = 0, 1, IITLE NO S *  ** 1	THE SHIP SIZE - 1 TAKELN - 1 LENGIN (HEADER(SNAME'INDEXS)) - 1 S	THE SHIP SIZE TAKGIN TO THE SHIP SHIP SHIP SHIP SHIP SHIP SHIP SHIP	ILLESM & "FOR I = 0, 1, TILLE"NO & "  TALLESM & "FOR I = 0, 1, TILLE"NO & "  THE SM & "FOR I = 0,	ITLE SW SFOR I = 0, 1, TITLE NO S  TEREST SW SFOR I = 0, 1, TITLE NO S  THE SW S	TILE SM &FOR I = 0, 1, TILE NO &  FREE SM &FOR I = 0, 1, TILE NO	THE SAME BY THE TARGEN TO THE
*NAKE BYTE = MAKGIN +  * LENGTH(HEADER(SNAME'INDEXS))+  * + 1 S  **********************************	**AKE**BYTE = MAKGIN +  * LENGTH(HEADER(SNAME*INDEX*))+  * TATE  **********************************	LENGTHINEADER(SNAME'INDEXS))  + 1 S  + 1 S  + 2 S  + 1 S	TLE SM S GARGE SMAGIN +  * LENGTHHEADER(SNAME'INDEXS))  * + 1	LENGTHHEADER(SNAME'INDEXS))  + 1	FRANKE BATE = MARGIN +  F. LENGHHEADER(SNAME'INDEXS))  F. 1. S.	LENGTHHEADER(SNAME'INDEXS))  + 1	LENGTHHEADER(SNAME'INDEXS))  + 1	LENGTHINEADER(SNAME INDEXS))  + 1	LENGTHINE ANGERIANGE INDEXS))  + 1 8  - 1 8  - 1 9	LENGTH (HEADER(SNAME INDEXS))  + 1	TILE'SM & "FOR I = "ARGIN +  * 1	TILE'SM \$FOR I = 0, 1, TITLE'NO \$  FILE'SM \$FOR I =	LENGTHIHEADER(SNAHE'INDEXS))  + 1	TILE'SH & "FOR I = "ARGIN +  + 1 &
* LENGTH(HEADER(SNAME*INDEXS))*  * + 1 S  **********************************	* LENGTH(HEADER(SNAME'INDEXS))*  * 1	* LENGTH(HEADER(SNAME INDEXS))  * 1	ILLE'SM & "FOR I = 0, 1, TITLE'NO & "  " LENGTH(HEADER(SNAME.INDEXS))"  " LENGTH(HEADER(SNAME.INDEXS))"  " LENGTH(HEADER(SNAME.INDEXS))"  " LENGTH STATE OF THE STATE OF	LENGTH(HEADER(SNAME INDEXS))  1 LENGTH(HEADER(SNAME INDEXS))  1 LESS S S S S S S S S S S S S S S S S S	ITLE'SM & "FOR I = 0, 1, TITLE'NO & "  FREE STATES OF THE STATES O	LENGTH(HEADER(SNAME INDEXS))  1 LENGTH(HEADER(SNAME INDEXS))  1 LESS S S S S S S S S S S S S S S S S S	LENGTH(HEADER(SNAME INDEXS))  1 LENGTH(HEADER(SNAME INDEXS))  1 LESS STATES STA	LENGTH(HEADER(SNAME'INDEXS))  LENGTH(HEADER(SNAME'INDEXS))  LILE'SH & "FOR I = 0, 1, TITLE'NO & "  THE SH	TLE'SM & "FOR I = 0, 1, TILE'NO & "  I = 0, 1, OUT'BUF'SIZE - 1 "  " LENGTH(HEADER(SNAME'INDEXS))  " LENGTH(HEADER(SNAME'INDEXS))  " LENGTH STATE   1   1   1   1   1   1   1   1   1	TILE'SM & "FOR I = 0, 1, TILE'NO & "  ITLE'SM & "FOR I = 0,	TILE'SM \$ "FOR I = 0, 1, TILE'NO \$ "  ITLE'SM \$ "FOR I = 0, 1, TILE'SM \$ "  ITLE'SM \$ "FOR I = 0, 1, TILE'SM \$ "  ITLE'SM \$ "FOR I = 0, 1, TILE'SM \$ "  ITLE'SM \$ "FOR I = 0, 1, TILE'SM \$ "  ITLE'SM \$ "FOR I = 0, 1, TILE'SM \$ "  ITLE'SM \$ "FOR I = 0, 1, TILE'SM \$ "  ITLE'SM \$ "FOR I = 0, 1, TILE'SM \$ "  ITLE'SM \$ "FOR I = 0, 1, TILE'SM \$ "  ITLE'SM \$ "FOR I = 0, 1, TILE'SM \$ "  ITLE'SM \$ "FOR I = 0, 1, TILE'SM \$ "  ITLE'SM \$ "FOR I = 0, 1, TILE'SM \$ "  ITLE'SM \$ "FOR I = 0, 1, TILE'SM \$ "  ITLE'SM \$ "FOR I = 0, 1, TILE'SM \$ "  ITLE'SM \$ "FOR I = 0, 1, TILE'SM \$ "  ITLE'SM \$ "FOR I = 0, 1, TILE'SM \$	LENGTH (HEADER(SNAME.INDEXS))  LENGTH (HEADER(SNAME.INDEXS))  LILE'SN SFOR I = 0, 1, TILLE'NO S "  PERTONENT STREET STRE	TILESH & "FOR I = 0, 1, TILE'NO & "  ITLE'SH & "FOR I = 0, 1	TILE'SM \$ "FOR I = 0, 1, TILE'NO \$ "FOR I =
*NAKE BYTE = MAKGIN + * LENGTH(HEADER(SNAME*INDEXS))+ * + 1 S	**ARE BYTE = MAKGIN +  * LENGTH(HEADER(SNAME*INDEXS))+  * + 1 \$  **********************************	LENGTH (HEADER(SNAME "INDEXS))  LENGTH (HEADER(SNAME "INDEXS))  LILE'SM S	LENGTH(HEADER(SNAME'INDEXS))  + 1	LENGTH(HEADER(SNAME"INDEXS))  1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	LENGTH(HEADER(SNAME'INDEXS))  + 1  + 1  - 1	LENGTH(HEADER(SNAME"INDEXS))  1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	LENGTH(HEADER(SNAME"INDEXS))  - 1	LENGTH (HEADER(SNAME'INDEXS))  LENGTH (HEADER(SNAME'INDEXS))  LILE'SM SFOR I = 0, 1, TITLE'NO S  LILE'SM S	LENGTH (HEADER(SNAME"INDEXS))  + 1	LENGTH (HEADER(SNAME"INDEXS))  + 1	LENGTH(HEADER(SNAME'INDEXS))  + 1	LENGTH(HEADER(SNAME"INDEXS))  + 1	LENGTH (HEADER(SNAME"INDEXS))  LENGTH (HEADER(SNAME"INDEXS))  LILE'SM S	ILESSM & "FOR I = 0, 1, TILE'NO & "  ILESSM & "FOR I = 0, 1, TILE'NO & "  ILESSM & "FOR I = 0, 1, TILE'NO & "FO
** LENGTH(HEADER(SNAME'INDEX*))*  * 1	**AKE*BYTE = MAKGIN + ** LENGTH(HEADER(\$NAME*INDEX\$))+ ***********************************	ITLE'SH & "FOR I = 0, 1, IITLE'NO \$ "  ITLE'SH & "FOR I = 0, 1, IITLE'NO \$ "  ITLE'SH & "FOR I = 0, 1, IITLE'NO \$ "  ITLE'SH & "FOR I = 0, 1, IITLE'NO \$ "  ITLE'SH & "FOR I = 0, 1, IITLE'NO \$ "  ITTLE'NO \$ "FOR I = 0, 1, IITLE'NO \$ "  ITTLE'NO \$ "FOR I = 0, 1, IITLE'NO \$ "  ITTLE'NO \$ "FOR I = 0, 1, IITLE'NO \$ "  ITTLE'NO \$ "FOR I = 0, 1, IITLE'NO \$ "  ITTLE'NO \$ "FOR I = 0, 1, IITLE'NO \$ "  ITTLE'NO \$ "FOR I = 0, 1, IITLE'NO \$ "  ITTLE'NO \$ "FOR I = 0, 1, IITLE'NO \$ "  ITTLE'NO \$ "FOR I = 0, 1, IITLE'NO \$ "  ITTLE'NO \$ "FOR I = 0, 1, IITLE'NO \$ "  ITTLE'NO \$ "FOR I = 0, 1, IITLE'NO \$ "  ITTLE'NO \$ "FOR I = 0, 1, IITLE'NO \$ "  ITTLE'NO \$ "FOR I = 0, 1, IITLE'NO \$ "  ITTLE'NO \$ "FOR I = 0, 1, IITLE'NO \$ "  ITTLE'NO \$ "FOR I = 0, 1, IITLE'NO \$ "  ITTLE'NO \$ "FOR I = 0, 1, IITLE'NO \$ "  ITTLE'NO \$ "FOR I = 0, IITLE'NO \$ "  ITTLE'NO \$ "FO	LENGTH(HEADER(SNAHE'INDEXS))*  * 1	ILLE'SM 8 "FOR I = 0, 1, TILLE'NO 8 "  ILLE'SM 8 "	LENGTH (HEADER(SNAME INDEXS)) + 1	ILLE'SW & "FOR I = MAKGIN +  1 A A A A A A A A A A A A A A A A A A	ILLE'SM 8 "FOR I = 0, 1, TILLE'NO 8 "  I	ITLE'SM \$ "FOR I = 0, 1, TITLE'NO \$ "  I = 0, 1, OUT BUF'SIZE - 1 "OUT LINE(SIZE)  *** *** *** *** *** *** *** *** *** *	LENGTH(HEADER(SNAME INDEXS)) + 1 S	LENGTHHEADER(SNAME INDEXS))  LENGTHHEADER(SNAME INDEXS))  LILE'SN SFOR I = 0, 1, TILLE'NO S  ITLE'SN S	LENGTH(HEADER(SNAME'INDEXS))  LENGTH(HEADER(SNAME'INDEXS))  LA S  LENGTH(HEADER(SNAME'INDEXS))  LILE'SN S  LENGTH(HEADER(SNAME'INDEXS))  LILE'SN S  LENGTH(HEADER(SNAME'INDEXS))  LILE'SN S  LENGTH(HEADER(SNAME'INDEXS))  LENGTH(HEADER(SNAME'INDEXS))  LENGTH(HEADER(SNAME'INDEXS))  LENGTH(HEADER(SNAME'INDEXS))  LENGTH(HEADER(SNAME'INDEXS))	TLESS STEE HARGIN +  LENGTH (HEADER(SNAHE' INDEXS)) +  LENGTH (HEADER(SNAHE' INDEXS)) +  LESS S S S S S S S S S S S S S S S S S S	ITLE'SM \$ "FOR I = 0, 1, IILE'NO \$ "  I = 0, 1, OUT-BUF'SIZE - 1 "OUT-LINE(SIZE)  **	LENGTH HEADER(SNAME INDEXS))  LENGTH HEADER(SNAME INDEXS))  LESS SFOR I = 0, 1, TILLE NO S  ITLE SN S -
**************************************	**AAKE*BYTE = MAKGIN +  * LENGH(HEADER(SNAME*INDEXS))+  * + 1	**************************************	LENGTH (HEADER(SNAME. INDEXS))  LENGTH (HEADER(SNAME. INDEXS))  LA 1	*** **********************************	TLE SW SFOR I = 0, 1, TILE NO SFOR I = 0	*** **********************************	*** **********************************	*** **********************************	* LENGTH(HEADER(SNAME'INDEXS))	*** **********************************	TLESS SFOR I = 0, 1, TILE'NO SFOR I = 0,	** ** ** ** ** ** ** ** ** ** ** ** **	* LENGTH(HEADER(SNAME'INDEXS)) * LA S * TENGTH(HEADER(SNAME'INDEXS)) * LA S * TENGTH (HEADER(SNAME'INDEXS)) * LA S * TENGTH (HEADER(SNAME'INDEXS) * TENGTH (HEADER	** ** ** ** ** ** ** ** ** ** ** ** **
* S * NAKE BYTE = MAKGIN + * LENGTH(HEADER(SNAME'INDEXS)) + 1 S * * * * * * * * * * * * * * * * * *	* AAKE BYTE = MAKGIN + * LENGTH(HEADER(SNAME'INDEXS))* * 1	TENGRAPH E MARGIN +  * LENGRICH HEADER(SNAME INDEXS))  * + 1 S  *	**AME**BYTE = MAKGIN + ** LENGHHHEADER(SNAME*INDEXS)); ** + 1	** AAKE BYTE = MAKGIN +  ** LENGTHHEADER(SNAME'INDEXS))*  ** + 1	**AME**BYTE = MAKGIN + ** LENGTH(HEADER(SNAME*INDEXS)); ** + 1	** AAKE BYTE = MAKGIN +  ** LENGTHHEADER(SNAME'INDEXS))*  ** + 1	TLE SM SFOR I = 0, 1, TILE NO S  ITLE SM S	THE SAME BYTE = MAKGIN +  " LENGTHHEADER(SNAME INDEXS))  " 1	THE SWARE BYTE = MAKGIN +  " LENGTHHEADER(SNAME INDEXS))  " 1	THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S  THE SH SFOR I S O 1 TILE NO S O 1 T	TILE'SM & "FOR I = 0, 1, TILE'NO & "  I = 0, 1, OUT'BUF'SIZE - 1 * "OUT'LINE(SIZE)	** XAKE BYTE = MAKGIN +  ** LENGTH(HEADER(SNAME'INDEXS)) +  ** + 1	TENGRAPE = MARGIN +  * LENGTHHEADER(SNAME'INDEXS))  * 1	LENGTH(HEADER(SNAME'INDEXS))  1 LENGTH(HEADER(SNAME'INDEXS))  1 LESSW SFOR I = 0, 1, TITLE'NO S  1 LESSW SFOR I = 0, 1, TITLE'NO S  1 LE 0, 1, OUT BUF SIZE - 1 * ********************************
* SENGTH (HEADER(SNAME*INDEXS)) + LENGTH (HEADER(SNAME*INDEXS)) + 1 S +	* S * S * S * S * S * S * S * S * S * S	* S * NAKE BYTE = MAKGIN + * LENGTH (HEADER(SMAME 'INDEXS)) + * 1 \$ * * * * * * * * * * * * * * * * *	** AAKE BYTE = MAKGIN +  ** LENGTH(HEADER(SNAME.INDEXS))*  ** 1	SANGE BYTE = MAKGIN +  * LENGTH (HEADER(SNAME 'INDEXS))  * LENGTH (HEADER(SNAME 'INDEXS)  * LENGTH (HEADER(SNAME	ITLE'SM & ***********************************	SANGE BYTE = MAKGIN +  * LENGTH(HEADER(SNAME INDEXS))  * 1	SANGE BYTE = MAKGIN +  * LENGTH (HEADER (SMAME 'INDEXS))  * 1 ENGTH (HEADER 'INDEXS)  * 1 ENGH	SANGE BYTE = MAKGIN +  * LENGTH (HEADER (SNAME 'INDEXS))  * 1	TLESS SFOR I = 0, 1, TILE'NO S  ILE 0, 1, OUT BUF'SIZE - 1	TILE'SM & "FOR I = 0, 1, TILE'NO & "  ITLE'SM & "FOR I = 0,	TILE'SH & "FOR I = 0, 1, TILE'NO & "  ENGTH(HEADER(SNAME'INDEXS))  * 1	SAME BYTE = MAKGIN +  * LENGTH(HEADER(SMAHE 'INDEXS))  * 1	SAME BYTE = MAKGIN +  * LENGTH (HEADER(SNAME 'INDEXS))  * 1 \$  *	TLE SM SFOR I = 0, 1, TILE NO S  I = 0, 1, OUT BUF SIZE - 1 * ********************************
* 58)(HEADER(SGS)) = 5H(PAGE)* * AKE BYTE = MAKGIN + * LENGTH(HEADER(SNAME*INDEX*))* * + 1	* 55)(HEADER(\$65)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME*INDEX\$))*  * + 1 \$  ** * 1 \$  ** * * * * * * * * * * * * * * * * *	* \$5)(HEADER(\$6\$)) = \$H(AGE) * *NAKE*BYTE = HAKGIN * * LENGTH(HEADER(\$NAME*INDEX\$))* * 1	* 55)(HEADER(\$6\$)) = 5H(PAGE)*  * A K* * B YE = MAKGIN +  * LENGTH(HEADER(\$NAME*INDEX\$))*  * + 1	TELESH STEE HARGIN +  * LENGTH(HEADER(SNAME 'INDEXS)) +  * LENGTH(HEADER(SNAME 'INDEXS) +  * LENGTH(HEADER(SNAME 'INDEXS)) +  * LENGTH(HEADER(SNAME 'INDEXS) +  * LENGTH(HEADER(SNAME 'INDEXS)) +  * LENGTH(HEADER(SNAME 'INDEXS) +  * LENGTH(HEADER(SNAME 'INDEXS)) +  * LENGTH(HEADER(SNAME 'INDEXS) +  * LENGTH(HEADER(SNAME '	THE SWARE BYTE = MAKGIN +  ** LENGTH(HEADER(SNAME*INDEXS))*  ** 1	THE SWEET HANGIN +  * LENGTH(HEADER(SNAME 'INDEXS)) +  * LENGTH(HEADER(SNAME 'INDEXS) +  * LENGTH(HEADER(SNAME 'INDEXS)) +  * LENGTH(HEADER(SNAME 'INDEXS) +  * LENGTH(HEADER(SNAME '	THE SAME SAME SAME SAME SAME SAME SAME SAM	TILE'SM & "FOR I = 0, 1, TILE'NO & "  ITLE'SM & "FOR I = 0, 1, TILE'SM & "  ITLE'SM & "FOR I =	* \$5)(HEADER(\$6\$)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME*INDEX\$))*  * + 1	* \$5)(HEADER(\$6\$)) = \$H(PAGE)*  * NAME*BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME*INDEX\$))*  * 1 \$  * * 1 \$  * * 1 \$  * * * 1 \$  * * * * *  * * * * *  * * * * *  * * * * *  * * * * *  * * * * *  * * * * *  * * * * *  * * * * *  * * * * *  * * * * * *  * * * * * * *  *	* \$5)(HEADER(\$6\$)) = 5H(PAGE)*  * * * * * * * * * * * * * * * * * *	* 55)(HEADER(\$65)) = 5H(PAGE)*  * A & * NAKE* BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME*INDEX\$))*  * 1 & * 1 & * * * * * * * * * * * * * *	* \$5)(HEADER(\$6\$)) = 5H(PAGE)*  *NAKE*BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME*INDEX\$))*  * + 1	* \$5)(HEADER(\$6\$)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME*INDEX\$))*  * + 1 \$  * * + 1 \$  * * + 1 \$  * * + 1 \$  * * * * * * * * * * * * * * * * * *
+ 58) (HEADER(\$68)) = 5H(PAGE)+  * NAKE 8 YE = MAKGIN +  * LENGTH (HEADER(\$NAME'INDEX\$))+  * 1	* 58) (HEADER(\$6\$)) = 5H(PAGE)*  *NAKE*BYTE = MAKGIN *  * LENGTH(HEADER(\$NAME*INDEX\$))*  * 1	* \$\$)(HEADER(\$6\$)) = 5H(PAGE)*  *NAKE*BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME*INDEX\$))*  * 1	TLE'SW ************************************	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$FOR I = 0, 1, TIT	ILLE'SW ************************************	ITLE'SM \$FOR I = 0, 1, TITLE'NO \$FOR I = 0, 1, TITLE'SW \$FOR I = 0, 1, TITLE'NO \$FOR I = 0, 1, TIT	ITLE'SM \$ "FOR I = 0, 1, TITLE'NO \$ "  ITLE'SM \$ "FOR I =	FSS (HEADER(\$63)) = 5H(PAGE) = \$  **NAKE*BYTE = MAKGIN +   ** LENGTH (HEADER(\$NAME*INDEX\$)) =   ** L	FSS)(HEADER(\$63)) = 5H(PAGE)*  **NAHE*BYTE = MAKGIN +  **LENGTH(HEADER(\$NAHE*INDEX\$))*  **1.5.  **LENGTH(HEADER(\$NAHE*INDEX\$))*  **1.5.  **LENGTH(HEADER(\$NAHE*INDEX\$))*  **1.5.  **LENGTH(HEADER(\$NAHE*INDEX\$))*  **LENGTH(HEADER(\$NAH	* \$\$)(HEADER(\$6\$)) = 5H(PAGE)*  **AENGTH(HEADER(\$NAME 'INDEX\$))*  * 1	* \$\$)(HEADER(\$6\$)) = 5H(PAGE)*  **AKE********************************	ILLE'SM \$ "FOR I = 0, 1, TILLE'NO \$ "  ILLE'SM \$ "FOR I = 0, 1, TILLE'NO \$ "	FERRET SERVICE STANDER SERVICE	* \$\$)(HEADER(\$6\$)) = 5H(PAGE)*  **ANE'E BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME'INDEX\$))*  * 1
+ 58)(HEADER(868)) = 5H(PAGE ) + 8	* 55)(HEADER(\$6\$)) = 5H(PAGE)*  **NAKE*BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME*INDEX\$))*  * 1 \$ *******************************	SS) (HEADER(SGS)) = 5H(PAGE) * NAKE BYTE = MAKGIN + * LENGTH(HEADER(SNAME'INDEXS)) * + 1 S * * * 1 S * * * * * * * * * * * * * * * * * * *	SS) (HEADER(SUS) = 5H(PAGE) = 5S) (HAGE) = 5S) (HADER(SUS) = 5S) (	SS)(HEADER(SUB) = 5H(PAGE)  ***AME**BYTE = MAKGIN +  ***LENGTH(HEADER(SNAME*INDEXS))  **********************************	TILE'SM SFOR I = 0, 1, TILE'NO S  ITLE 0, 1, OUT BUF'S IZE - 1	SS)(HEADER(SUB) = 5H(PAGE)  ***AME**BYTE = MAKGIN +  ***LENGHHHEADER(SNAME*INDEXS))  **********************************	SS)(HEADER(SUB) = 5H(PAGE)  ***AAKE*BYTE = MAKGIN +  ** LENGTH(HEADER(SNAME*INDEXS))  ** + 1	SS)(HEADER(SGS)) = 5H(PAGE) **AAKE*BYTE = MAKGIN +  ** LENGTH(HEADER(SNAHE*INDEXS))  ** + 1	S (HEADER(S6S)) = 5H(PAGE) * NAKE BYTE = MAKGIN + * LENGTH(HEADER(SNAME'INDEXS)) * + 1	* S\$) (HEADER(\$6\$) = 5H(PAGE )  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME INDEX\$))  * 1 1 2 4 1 1 1 IILE NO 8 ****  ** ILESS # ******  ** * * * * * * * * * * * * *	ILLE*SW \$ **********************************	SS) (HEADER(SUS) = 5H(PAGE) = 5S) (HEADER(SUS) = 5H(PAGE) = 5S) (HEADER(SUS) = 5H(PAGE) = 5S) (HEADER(SUS) =	S S S S S S S S S S S S S S S S S S S	TILE'SH & "FOR I = 0, 1, DITLE'NO & "FOR I = 0, 1, DUT'BUF'SIZE - 1 " "FOR I = 0, 1, DITLE'SH & "FOR I = 0, 1, DITLE'SH
*BYTE(\$PAGE*MIDIM - 10. * \$\$ (HEADER(\$6\$)) = 5H(PAGE)***  * * * * * * * * * * * * * * * * *	*SYTE(\$PAGE WIDIM - 10.  * \$\$ (HEADER(\$G\$)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN *  * LENGTH(HEADER(\$NAME INDEX\$))*  * 1 \$  ******************************	* SYTE (\$PAGE WIDTH = 10.  * \$  * NAME BYTE = MAKGIN +  * LENGTH (HEADER (\$NAME 'INDEX\$))  * * 1 \$  * 1 \$  * 1 \$  * * 1 \$  * 1 \$	*SYTE(\$PAGE WIDIM - 10.  *\$  *NAKE BYTE = MAKGIN +  * LENGTH(HEADER(SNAME INDEXS))  * + 1	* BYTE (\$PAGE ** NOT** - 10**  * \$ ** ***AME**BYTE = MAKGIN + ** ***LENGTH(HEADER(\$NAHE**INDEX\$))* **********************************	*SYTE(\$PAGE WIDIM - 10.  *\$  *NAKE BYTE = MAKGIN +  * LENGTH(HEADER(SNAME INDEXS))  * 1	* BYTE (\$PAGE ** NOTA - 10.  * \$ ** ******************************	* BYTE (\$PAGE ** NOT** = 5H(PAGE )**  * \$ **AAKE** BYTE = MAKGIN **  * LENGTH(HEADER(\$NAHE** INDEX\$))**  * 1 \$ ******************************	* BYTE (\$PAGE ** NOT** - 10.  * \$ \$ (HEADER(\$6\$)) = 5H(PAGE ).  * * * * * * * * * * * * * * * * * * *	* BYTE (\$PAGE ** NOT** - 10.  * \$ \$ (HEADER(\$G\$)) = 5H(PAGE ).  * * * * * * * * * * * * * * * * * * *	* BYTE (\$AAGE WIDTH - 10.  * \$\$ (HEADER(\$G\$)) = 5H(PAGE).  * * * * * * * * * * * * * * * * * * *	#BYTE (\$AAGE WIDTH = 10.	* BYTE (\$AAGE WIDTH = 10.  * \$ \$ (HEADER(\$G\$)) = 5H(PAGE)    * * * * * * * * * * * * * * * * * *	* BYTE (\$AAGE WIDIN - 10.  * \$\$ (HEADER(\$G\$)) = 5H(PAGE)*  * * * * * * * * * * * * * * * * * *	* BYTE (\$AAGE WIDTH - 10.  * \$ \$ (HEADER(\$G\$)) = 5H(PAGE).  * * * * * * * * * * * * * * * * * * *
* SY) (HEADER(SGS)) = 5H(PAGE)*  * SA) (HEADER(SGS)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN *  * LENGTH(HEADER(SNAME*INDEXS))*  * * 1 S  ****************************	* SY) (HEADER(\$6\$)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME'INDEX\$))*  * 1	* S\$) (HEADER(\$6\$)) = 5H(PAGE)**  **SAHE**BYTE = MAKGIN +  ** LENGTH(HEADER(\$NAME*INDEX\$))**  ** LENGTH(HEADER(\$NAME*INDEX\$)**  ** LENGTH(HEADER(\$NAME*INDEX\$))**  ** LENGTH(HEADER(\$NAME*INDEX\$)**  **	* SS)(HEADER(SGS)) = 5H(PAGE)*  * SA)(HEADER(SGS)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(SNAME*INDEXS))*  * + 1	* \$\$ (HEADER(\$6\$)) = 5H(PAGE)**  * \$\$ (HEADER(\$6\$)) = 5H(PAGE)**  * * * * * * * * * * * * * * * * * *	* S\$)(HEADER(\$6\$)) = 5H(PAGE)**  ***AKE**BYTE = MAKGIN +  ***LENGTH(HEADER(\$NAME*INDEX\$))**  *** 1	* \$\$ (HEADER(\$6\$)) = 5H(PAGE)**  * \$\$ (HEADER(\$6\$)) = 5H(PAGE)**  * * * * * * * * * * * * * * * * * *	SS (HEADER(SES) = 5HPAGE)  SS (HEADER(SES)) = 5HPAGE)  **NAME**BYTE = MAKGIN +  **LENGTH(HEADER(SNAME**INDEXS))  **********************************	SA) (HEADER(\$6\$)) = 5H(PAGE)  SA) (HEADER(\$6\$)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(\$NAME. INDEX\$))  * 1	* S\$)(HEADER(\$&\$)) = 5H(PAGE)*  * S\$)(HEADER(\$&\$)) = 5H(PAGE)*  * NAME BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME.INDEX\$))*  * 1	SBYTE (SPAGE ** MIDTA - 10.  SS) (HEADER(\$6\$)) = 5H(PAGE)*  **NAKE BYTE = MAKGIN +  ** LENGTH (HEADER(\$NAME ** INDEX\$))*  ** + 1	* S\$)(HEADER(\$6\$)) = 5H(PAGE)*  * S\$)(HEADER(\$6\$)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME"INDEX\$))*  * + 1	* \$\$)(HEADER(\$&\$)) = \$H(PAGE)**  * \$\$)(HEADER(\$&\$)) = \$H(PAGE)**  * * * * * * * * * * * * * * * * * *	SA) (HEADER(\$6\$)) = 5H(PAGE) = 5H	SET (SEADER (SES) = 5H(PAGE) = 5H
* SYTE (SPAGE* WIDTH - 10.  * S\$ (HEADER(\$6\$)) = 5H(PAGE)*  * NAKE* BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME*INDEX*))*  * 1	* BYTE (\$PAGE*MIOTH - 10.  * 58) (HEADER(\$68)) = 5H(PAGE)*  * NAKE*BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME*INDEX\$))*  * 1	* SYTE(SPAGE WIOTH - 10.  * SS)(HEADER(SGS)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(SNAME INDEXS))*  * 1 S  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(SNAME INDEXS))*  * 1 S  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(SNAME INDEXS))*  * 1 S  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(SNAME INDEXS))*  * 1 S  * NAKE BYTE = 1 TILLE NO	* S\$)(HEADER(\$6\$)) = 5H(PAGE)*  * S\$ ********************************	**SYTE(\$PAGE**MIDTH - 10.  * 58)(HEADER(\$6\$)) = 5H(PAGE)*  **NAKE**BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME**INDEX\$))*  * 1	* S\$)(HEADER(\$6\$)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAHE'INDEX\$))*  * 1	**SYTE(\$PAGE**MIDTH - 10.  * 58)(HEADER(\$G\$)) = 5H(PAGE)*  **NAKE**BYTE = MAKGIN +  **LENGTH(HEADER(\$NAME**INDEX\$))*  * 1	**SYTE(\$PAGE**MIDTH - 10.  * 58)(HEADER(\$6\$)) = 5H(PAGE)**  **NAKE**BYTE = MAKGIN +  **LENGTH(HEADER(\$NAME**INDEX\$))**  * 1	* SYTE(\$PAGE WIOTH - 10.  * SS)(HEADER(\$G\$)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME INDEX\$))*  * 1 \$	* S\$)(HEADER(\$6\$)) = 5H(PAGE)**  * S\$)(HEADER(\$6\$)) = 5H(PAGE)**  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME"INDEX\$))**  * + 1	* S\$)(HEADER(\$6\$)) = 5H(PAGE)*  * S\$)(HEADER(\$6\$)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME"INDEX\$))*  * 1	* S\$)(HEADER(\$6\$)) = 5H(PAGE)*  * S\$ ********************************	* SYTE(SPAGE * MIOTH - 10.  * SS)(HEADER(SGS)) = 5H(PAGE)*  * NAKE * BYTE = MAKGIN +  * LENGTH(HEADER(SNAME 'INDEXS))*  * 1	* S\$)(HEADER(\$6\$)) = 5H(PAGE)*  * S\$)(HEADER(\$6\$)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME INDEX\$))*  * 1	* S\$)(HEADER(\$6\$)) = 5H(PAGE)*  **********************************
*BYTE(\$PAGE*MIDTH - 10.  5 \$) (HEADER(\$6\$)) = 5H(PAGE)*  * NAKE*BYTE = MAKGIN *  * LENGTH (HEADER(\$NAME*INDEX\$))*  * 1 \$	**************************************	SSYF(SPAGE'MIDTH - 10.  SSYNHEADER(SUS) = 5H(PAGE)  NAME'BYTE = MAKGIN +  LENGTH(HEADER(SNAME'INDEXS))  + 1	### ### ##############################	ENTE(SPAGE MIDTH = 10.  5 \$1 (HEADER(SGS)) = 5H(PAGE)  *NAHE BYTE = MAKGIN +  *LENGTH(HEADER(SNAHE INDEXS))  * 1 \$  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ **FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ **	TLE SW STEE SPACE WIDTH - 10.  5 \$1 (HEADER (\$6.8.)) = 5H(PACE)  * NAME BYTE = MAKGIN +  * LENGTH (HEADER (\$NAME : INDEX\$))  * 1 \$  ITLE SW \$ **FOR I = 0, 1, TITLE **NO \$ **  ITLE **NOUT **SW SW S	EBYTE (SPAGE MIDTH - 10.  5 \$1 (HEADER (\$6\$)) = 5H(PAGE)  *NAHE BYTE = MAKGIN +  *LENGTH (HEADER (\$NAME INDEX\$))  * 1 \$  ITLE SN \$ *FOR I = 0, 1, TITLE NO \$ *  ITLE SN \$ *FOR I = 0, 1, TITLE	ENTE(SPAGE MIDTH - 10.  5 \$1 (HEADER(SGS)) = 5H(PAGE)  *NAHE BYTE = MAKGIN +  *LENGTH(HEADER(SNAHE INDEXS))  * 1 \$  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, 1, TITLE NO \$ **	ENTE (SPAGE MIDTH - 10.  5 \$1 (HEADER(\$6\$)) = 5H(PAGE)  8 NAKE BYTE = MAKGIN +  1 LENGTH (HEADER(\$NAME INDEX\$))  4 1 \$  1 1	SYTE(SPAGE WIDTH - 10.  SS (HEADER(SES)) = 5H(PAGE)  NAME BYTE = MARGIN +  LENGTH(HEADER(SNAME INDEXS))  + 1	SYTE(SPAGE ** LOT* - 10.  S\$ (HEADER(S&S)) = 5H(PAGE)  **NAKE BYTE = MAKGIN +  **LENGHHHEADER(SNAME INDEXS))  **1	TLESS SFOR I = 0, 1, TITLE'NO SFOR I = 0, 1, TITLE	SYTE(SPAGE MIDTH - 10.  5 \$1 (HEADER(SGS)) = 5H(PAGE)  * NAHE BYTE = MAKGIN +  * LENGTH (HEADER(SNAHE INDEXS))  * 1 \$  ITLE SN \$ *FOR I = 0, 1, ITILE NO \$ *  ITLE SN \$ *FOR I = 0, 1, ITILE N	SYTE(SPAGE WIDTH - 10.  SS (HEADER(SES)) = 5H(PAGE)  NAME BYTE = MARGIN +  LENGTH(HEADER(SNAME INDEXS))  + 1	### ### ##############################
**************************************	**SYTE(\$PAGE WIDTH - 10.  * 5\$) (HEADER(\$G\$)) = 5H(PAGE )**  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME 'INDEX\$))**  * 1 \$  ** 1 \$  *** 1 \$  ** 1 \$  ** 1 \$  *** 1 \$  *** 1 \$  *** 1 \$  *** 1 \$  *** 1 \$  *** 1 \$  *** 1 \$  *** 1 \$  *** 1 \$  *** 1 \$  *** 1 \$  *** 1 \$  *** 1 \$  *** 1	**************************************	**************************************	### STEE SPACE WIDTH - 10.  ### SS (HEADER(\$6.5)) = 5H(PAGE)  ### SS (HEADER(\$6.5)) = 5H(PAGE)  ### LENGTHHEADER(\$NAHE'INDEX\$))  #### 1	### STATE (SPACE WIDTH - 10.  * \$ \$ \$ (HEADER(\$6.8)) = 5H(PAGE) = 5 **  * * * * * * * * * * * * * * * * *	### STEE SPACE WIDTH - 10.  ### STEE SPACE WIDTH - 10.  ### STEE SPACE STEEN **  ### STEEN STEE SPACE STEEN	SYTE (SPACE WIDTH - 10.  SS) (HEADER(SGS)) = 5H(PAGE)  **AAKE BYTE = MAKGIN +  ** LENGTHHEADER(SNAME'INDEXS))  ** 1	### STEE SPACE WIDTH - 10.  • 58) (HEADER(\$6.)) = 5H(PAGE)  • 8 **********************************	### ##################################	**************************************	BYTE (\$AGE WIDTH - 10.  • 5\$) (HEADER(\$G\$)) = 5H(PAGE )  • NAKE BYTE = MAKGIN +  • LENGTH (HEADER(\$NAME 'INDEX\$))  • 1 \$	### STEE (SPACE WIDTH - 10.  * \$ \$ (HEADER(\$6.8)) = 5H(PAGE) = 5H(PAGE) = 5 (HEADER(\$6.8)) = 5H(PAGE) = 5 (HEADER(\$6.8)) = 5 (H	SYTE (SPACE WIDTH - 10.  SS) (HEADER(SGS)) = 5H(PAGE)  **NAKE BYTE = MAKGIN +  **LENGTH(HEADER(SNAHE'INDEXS))  **1	##YTE(\$PAGE WIDTH - 10.  * 5\$) (HEADER(\$G\$)) = 5H(PAGE ).  * NAME BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME INDEX\$)).  * 1 \$
* 58) (HEADER(\$68)) = 5H(PAGE)*  * 58) (HEADER(\$68)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME' INDEX*))*  * + 1 \$  **********************************	* SYTE (SPAGE * MIDTA - 10.  * SS) (HEADER(\$6\$)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME 'INDEXS))*  * 1 \$  * * 1 \$  * * * 1 \$  * * * * * * * * * * * * * * * * * *	BYTE (SPAGE MIDTA - 10,  5 \$ \$ (HEADER (\$6\$)) = 5H(PAGE)  8 **NAKE BYTE = MAKGIN +  **LENGTH (HEADER (\$NAME * INDEX\$))  **********************************	* SAYTE (SPAGE*WIDTH - 10.  * SS) (HEADER(\$6\$)) = 5H(PAGE)*  * NAKE*BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME*INDEX\$))*  * 1	BYTE (\$PAGE MIDTA = 10,  \$ \$ (HEADER(\$6\$)) = 5H(PAGE)  **NAKE BYTE = MAKGIN +  **LENGTH(HEADER(\$NAME INDEX\$))  **1 \$  ***TEE'SW \$ ***********************************	BYTE (SPAGE WIDTH - 10.  5 \$3 (HEADER(\$6\$)) = 5H(PAGE)  8 NAKE BYTE = MAKGIN +  1 LENGTH(HEADER(\$NAME INDEXS))  4 1 8  1 1	BYTE (SPAGE WIDTH - 10,  5 5) (HEADER(\$63)) = 5H(PAGE)  5 NAKE BYTE = MAKGIN +  1 LENGTH(HEADER(\$NAME 'INDEX\$))  7 1	BYTE (\$PAGE MIDTA = 10,  \$ \$ (HEADER(\$6\$)) = 5H(PAGE)  **NAKE BYTE = MAKGIN +  **LENGTH(HEADER(\$NAME INDEX\$))  **1 \$  **TLE SW \$ **********************************	SAYTE (SPAGE MIDTA - 10.  5 \$ (HEADER (\$6.8)) = 5H(PAGE).  **NAME BYTE = MAKGIN +  **LENGTH (HEADER (\$MAME * INDEX\$)).  **1 \$  *	BYTE (SPAGE MIDTA - 10.  5 \$) (HEADER (\$6.8)) = 5H(PAGE).  **NAHE BYTE = MAKGIN +  **LENGTH (HEADER (\$NAHE * INDEXS)).  ***A ********************************	BYTE (SPAGE WIDTH - 10.  SS) (HEADER(\$6\$)) = 5H(PAGE)  SNAKE BYTE = MAKGIN +  LENGTH (HEADER(\$NAME INDEXS))  + 1	BYTE (SPAGE WIDTH - 10.  SS) (HEADER(SGS)) = 5H(PAGE)  NAKE BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))  + 1	BYTE (SPAGE WIDTH - 10,  5 5) (HEADER(\$63)) = 5H(PAGE)  5 NAKE BYTE = MAKGIN +  1 LENGTH(HEADER(\$NAME 'INDEX\$))  5 + 1 \$	SAYTE (SPAGE MIDTA - 10.  5 \$ \$ (HEADER (\$6.8)) = 5H(PAGE).  **NAME BYTE = MAKGIN +  **LENGTH (HEADER (\$MAME * INDEX\$)).  **1 \$	BYTE (SPAGE WIDTH - 10.  55) (HEADER(\$6\$)) = 5H(PAGE)  5 NAKE BYTE = MAKGIN +  1 LENGTH (HEADER(\$NAME INDEX\$))  7 1 \$  1 TLE SM \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0, 1, TITLE NO \$  1 TLE NO \$FOR I = 0
**************************************	**************************************	* SYTE(\$AGE*WIDTH - 10. * S\$)(HEADER(\$6\$)) = 5H(PAGE)* * NAKE*BYTE = MAKGIN + * LENGTH(HEADER(\$NAME*INDEX\$))* * 1 \$ * * 1 \$ * * * 1 \$ * * * * * * * * * * * * * * * * * * *	**************************************	*** SYTE(\$\$AGE***********************************	**************************************	*** SYTE(\$\$AGE***********************************	BYTE(SPAGE WIDTH - 10.  58) (HEADER(SGS)) = 5H(PAGE).  SAMÉE BYTE = MARGIN +  LENGTH (HEADER(SNAME INDEXS)).  1 S	### SECTION AND AND AND AND AND AND AND AND AND AN	**SYTE(SPACE**MIDTH - 10.  * 58)(HEADER(\$C\$)) = 5H(PAGE)**  **NAKE**BYTE = MAKGIN +  ** LENGTH(HEADER(\$NAME*INDEX\$))**  * 1	**************************************	**************************************	*** SYTE(\$\$AGE***MIDTH - 10.  *** S\$\$ (HEADER(\$\$B\$)) = \$H(PAGE)***  *** NAKE************************************	* S\$)(HEADER(\$\$)) = \$HIPAGE)*  * S\$)(HEADER(\$\$)) = \$HIPAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME'INDEX\$))*  * 1	**************************************
**************************************	**************************************	BYTE(SPAGE MIDTH - 10.  5 \$) (HEADER(\$6\$)) = 5H(PAGE)  8 NAKE BYTE = MAKGIN +  1 LENGTH(HEADER(\$NAME INDEX\$))  4 1 8  1 1	### STEE (SPAGE MIDTH = 10.  * \$\$ (HEADER(\$6\$)) = \$H(PAGE)*  * NAME BYTE = MARGIN +  * LENGTH (HEADER(\$NAME : NDEX\$))*  * 1 \$  *	### STATE CONTROL TO THE TOTAL THE	ENTE(SPAGE MIDTH = 10.  5 \$) (HEADER(S&S)) = 5H(PAGE)  5 \$  *NAHE BYTE = MARGIN +  *LENGTH (HEADER(SNAHE INDEXS))  * 1 \$  ITLE*SN \$ **********************************	### STATE CONTRACTOR TO THE STATE OF THE STA	### STATE CONTROL OF THE CONTROL OF	BYTE(SPAGE MIDTH - 10.  5 \$) (HEADER(\$6\$)) = 5H(PAGE)  8 ***AME**BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME**INDEX\$))  * + 1 \$  ****ENGTHER THE STATE = NATHER S	BYTE(SPAGE MIDTH - 10.  5 \$) (HEADER(S&S)) = 5H(PAGE)  8 NAKE BYTE = MAKGIN +  1 LENGTH(HEADER(SNAME INDEXS))  4 1 8  1	**************************************	### STEE (SPAGE MIDTH = 10.  * \$\$ (HEADER(S&S)) = 5H(PAGE)*  * * * * * * * * * * * * * * * * * *	### STATE CONTRACTOR TO THE ST	BYTE(SPAGE MIDTH - 10.  5 \$) (HEADER(S&S)) = 5H(PAGE)  8 NAKE BYTE = MAKGIN +  1 LENGTH(HEADER(SNAME INDEXS))  4 1 8  1 LESS 8FOR I = 0, 1, TITLE NO 8  ITLE SN 8FOR I = 0, TITLE NO 8  ITLE SN 8FOR I = 0, TITLE NO 8  ITLE SN 8FOR I = 0, TITLE NO 8  ITLE SN 8FOR I = 0, TITLE NO 8  ITLE SN 8FOR I = 0, TITLE NO 8  ITLE SN 8FOR I = 0, TITLE NO 8  ITLE SN 8FOR I = 0, TITLE NO 8  ITLE SN 8FOR I = 0, TITLE NO 8  ITLE SN 8FOR I = 0, TITLE NO 8  ITLE SN 8FOR I = 0, TITLE NO 8  ITLE SN 8FOR I = 0, TITLE NO 8  ITLE SN 8FOR I = 0, TITLE NO 8  ITLE SN 8FOR I = 0, TITLE NO 8  ITLE SN 8FOR I = 0, TITLE NO 8  ITLE SN 8FOR I = 0, TITLE NO	**************************************
**************************************	**************************************	**************************************	**************************************	######################################	### ##################################	##YTE(\$PAGE*WIDTH = 10.  * \$5)(HEADER(\$65)) = 5H(PAGE)*  * **AAKE*BYTE = MAKGIN +  * * LENGTH(HEADER(\$NAME*INDEX\$))*  * * 1	**************************************	**************************************	**************************************	**************************************	ENTE (SPAGE WIDTH - 10.  SS) (HEADER(SS)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))  LILE'SN S **********************************	##YTE(\$PAGE*WIDTH - 10.  * \$5)(HEADER(\$6\$)) = 5H(PAGE)*  * NAKE*BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME*INDEX\$))*  * + 1	**************************************	**************************************
* SYTE (\$PAGE * MIDTH - 10.  * 5\$) (HEADER(\$6\$)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME'INDEX\$))*  * + 1 \$  **********************************	**************************************	**************************************	**************************************	**************************************	**************************************	**************************************	**************************************	**************************************	BYTE (\$ADE " MIT - 10,  • 5\$) (HEADER(\$£)) = 5H(PAGE)  • NAME BYTE = MARGIN +  • LENGTH (HEADER(\$NAME " INDEX\$))  • 1 \$	**************************************	ENTE(SPACE MIDIA - 10.  • \$\$ (HEADER(\$£\$)) = 5H(PAGE)  • \$ **NAKE BYTE = MAKGIN +  • LENGTH (HEADER(\$NAME * INDEX\$))  • • 1 \$  • • 1 \$  • • 1 \$  • • 1 \$  • • 1 * **ILE**NO \$  • 1 * *	**************************************	**************************************	BYTE(\$PAGE WIDTH - 10.  * \$\$)(HEADER(\$&\$)) = 5H(PAGE).  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME INDEX\$)).  * 1 \$  * 1 \$  * 1 \$  * 1 \$  * 1 \$  * 1 \$  * 1 * * * * * * * * * * * * * * * * *
* SYTE (SPAGE* MIDTH - 10.  * SS) (HEADER(SGS)) = 5H(PAGE)*  * NAKE BYTE = MARGIN +  * LENGTH(HEADER(SNAME*INDEX*))*  * + 1 S  **********************************	* SYTE (SPAGE MIDTA - 10.  * 5\$) (HEADER(\$6\$)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH (HEADER(\$NAME 'INDEX\$))*  * 1 \$  * * 1 \$  * * * * * * * * * * * * * * * * * *	BYTE (SPAGE MIDTH = 10,  \$ \$ (HEADER(\$6.8)) = 5H(PAGE)  **NAKE BYTE = MARGIN +  **LENGTH(HEADER(\$NAME.INDEX\$))  **1 \$  **TLE SW \$ **********************************	BYTE (SPAGE WIDTH - 10.  SS) (HEADER(SGS)) = 5H(PAGE).  NAKE BYTE = MAKGIN +  LENGTH(HEADER(SNAME INDEXS)).  + 1 S  SSECTION OF STREET	ENTE(EPAGE MIDTH - 10.  • \$\$ (HEADER(\$&\$1) = 5H(PAGE).  • NAME BYTE = MARGIN +  • LENGTH(HEADER(\$NAME INDEX\$)).  • 1 \$  •	BYTE (SPAGE WIDTH - 10.  \$ 5\$) (HEADER(\$6\$)) = 5H(PAGE).  **NAKE BYTE = MAKGIN +  **LENGTH(HEADER(\$NAME INDEX\$)).  **1	BYTE (SPAGE MIDTH - 10.  • \$\$) (HEADER(\$&\$1) = 5H(PAGE).  • NAME BYTE = MARGIN +  • LENGTH (HEADER(\$MAME * INDEX\$)).  • 1 \$  • 1	ENTE(EPAGE MIDTH = 10.  • \$\$ (HEADER(\$&\$1) = 5H(PAGE).  • NAKE BYTE = MARGIN +  • LENGTH(HEADER(\$NAME INDEX\$)).  • 1 \$  •	ENTE(EPAGE MIDTH = 10,  \$ \$ (HEADER(\$6.8)) = 5H(PAGE).  **NAKE BYTE = MARGIN +  **LENGTH(HEADER(\$NAME.INDEX\$)).  **ILESS	ENTE(EPAGE MIDTA - 10.  \$ \$ (HEADER(\$6\$)) = 5H(PAGE).  **NAME BYTE = MAKGIN +  **LENGTH(HEADER(\$NAME.INDEX\$)).  **1 \$  **ILE*SN \$ ***FOR I = 0, 1, TITLE*NO \$ ***FOR I = 0, 1, OUT*BUF*SIZE - 1 **********************************	BYTE (SPAGE MIDTA - 10.  S S) (HEADER(\$6\$)) = 5H(PAGE)  **NAHE BYTE = MAKGIN +  **LENGTH (HEADER(\$NAHE * INDEX\$))  **+ 1 \$  **+ 1 \$  **********************************	BYTE (SPAGE WIDTH - 10.  SS) (HEADER(SGS)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))  + 1 S  TILE SN SFOR I = 0. 1, TILLE NO S  ITLE SN SFOR I = 0. 1, TILLE NO S -	BYTE (SPAGE WIDTH - 10.  5 \$) (HEADER(\$6\$)) = 5H(PAGE)  5 **AKE**********************************	ENTE(SPACE MIDTA = 10,  \$ \$ (HEADER(\$6\$)) = 5H(PAGE)  **NAME**BYTE = MAKGIN +  **LENGTH(HEADER(\$NAME**INDEX\$))  **1 \$  **ILE*SN \$ **********************************	BYTE (SPAGE MIDTA - 10.  5 \$) (HEADER(\$6\$)) = 5H(PAGE)  **NAKE BYTE = MAKGIN +  **LENGTH (HEADER(\$NAME INDEX\$))  **+1 \$  **TLE SN \$ **********************************
* SYTE (\$PAGE* WIDTH = 10.  * SS) (HEADER(\$G\$)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAHE*INDEX*))*  * 1 \$  ******************************	**************************************	BYTE (SPAGE MIDTA - 10.  5 \$) (HEADER(\$6\$)) = 5H(PAGE)  **NAHE BYTE = MAKGIN +  **LENGTH (HEADER(\$NAHE * INDEX\$))  **********************************	* SS)(HEADEE'WIDTH - 10.  * SS)(HEADER(\$6\$)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME'INDEX\$))*  * 1 \$  * * 1 \$  * * * * * * * * * * * * * * * * * *	ENTE (SPAGE MIDTA - 10.  • \$\$ (HEADER(\$6\$)) = 5H(PAGE)  • NAKE BYTE = MAKGIN +  • LENGTH (HEADER(\$NAME INDEX\$))  • 1 \$  •	BYTE (SPAGE WIDTH - 10.  5 5) (HEADER(\$6.8)) = 5H(PAGE)  5 NAKE BYTE = MAKGIN +  LENGTH(HEADER(\$NAME 'INDEXS))  4 1 5  5 TILE SN 8FOR I = 0. 1. TITLE 'NO 8 '  ITLE SN 8FOR I = 0. 1. TITLE 'NO 8 '  ITLE SN 8FOR I = 0. 1. TITLE 'NO 8 '  IT = 0. 1. OUT BUF SIZE - 1 * ********************************	# BYTE (\$PAGE ** MIDTH - 10.  * 58) (HEADER(\$68)) = 5H(PAGE)*  ** ** ** ** ** ** ** ** ** ** ** ** *	BYTE (SPAGE MIDTA - 10.  • \$\$ (HEADER(\$6\$)) = 5H(PAGE)  • NAKE BYTE = MAKGIN +  • LENGTH (HEADER(\$NAME INDEX\$))  • 1 \$  • 1 \$  • 1 \$  • 1 \$  • 1 * OUT*BUF*SIZE - 1  • 1 * OUT*BUF*SIZE - 1	ENTE(SPAGE MIDTA - 10.  5 \$) (HEADER(\$6\$)) = 5H(PAGE)  **NAME BYTE = MAKGIN +  **LENGTH(HEADER(\$NAME * INDEX\$))  **1 \$  **ILE*SN \$ ***FOR I = 0, 1, TITLE*NO \$ **  ITLE*SN \$ **	BYTE (SPAGE WIDTH - 10.  SS) (HEADER (SGS)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER (SNAME INDEXS))  + 1	BYTE (SPAGE WIDTH - 10.  SS) (HEADER(SGS)) = 5H(PAGE).  NAME BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS)).  + 1 S  **********************************	BYTE (SPAGE WIDTH - 10.  • 58) (HEADER(\$68)) = 5H(PAGE) • 58) (HEADER(\$88)) = 5H(PAGE) • 58) (HEADER(\$88)) = 5H(PAGE) • 58) (HEADER(\$88)) = 5H(PAGE) • 58) (HEADER(\$88) = 5H(PAGE) • 1	#BYTE(SPAGE'WIOTH - 10.  * 58) (HEADER(\$68)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME'INDEX\$))*  * 1	BYTE (SPAGE WIDTH - 10.  5 \$) (HEADER(\$6\$)) = 5H(PAGE)  **NAHE BYTE = MAKGIN +  **LENGTH (HEADER(\$NAHE * INDEX\$))  **1 \$  **ILE*SW \$ ***FOR I = 0, 1, TITLE*NO \$ **  **ILE*SW \$ ***FOR I = 0, 1, TITLE*NO \$ **  **ITLE*SW \$ **	BYTE (SPAGE WIOTH - 10.  SS) (HEADER(SGS)) = 5H(PAGE).  NAME BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS)).  + 1 S  **********************************
**************************************	* S\$)(HEADER(\$6\$)) = 5H(PAGE)*  * S\$)(HEADER(\$6\$)) = 5H(PAGE)*  * NAKE 8VTE = MAKGIN +  * LENGTH(HEADER(\$NAME'INDEX\$))*  * 1 \$  ** * 1 \$  ** * * * * * * * * * * * * * * * * *	* S\$)(HEADER(\$6.8) = 5H(PAGE)*  * \$\$)(HEADER(\$6.8)) = 5H(PAGE)*  * NAME BYTE = MARGIN +  * LENGTH(HEADER(\$NAME INDEX\$))*  * 1.8  * 1.8  * 1.	**************************************	**************************************	**************************************	**************************************	BYTE(SPAGE WIDTH - 10.  58) (HEADER(SGS)) = 5H(PAGE)  **AKE BYTE = MAKGIN +  ** LENGTH (HEADER(SNAME INDEXS))  ** 1	**************************************	**************************************	**************************************	**************************************	**************************************	**************************************	**************************************
**************************************	**************************************	BYTE(SPAGE MIDTH - 10.  5 \$) (HEADER(\$6\$)) = 5H(PAGE)  8 \$  *NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME INDEX\$))  * 1 \$  **TEE'SN \$  ***********************************	**************************************	### STATE CAPE MIDTH = 10.  * \$\$ (HEADER(\$6\$)) = \$H(PAGE)**  * * * * * * * * * * * * * * * * * *	TLE'SW SFOR I = 0, 1, TILE'NO S SFOR I = 0, 1, OUT'BUF'SIZE - 1	### STATE (SPAGE MIDTH = 10.  * \$\$ (HEADER(S&3)) = 5H(PAGE)**  * * * * * * * * * * * * * * * * * *	### STATE CONTRACTOR TO THE TOTAL TO THE TOTAL	ENTE(SPAGE MIDTH = 10.  5 \$) (HEADER(\$6\$)) = 5H(PAGE)  8 \$  **NAHE*BYTE = MAKGIN +  ** LENGTH(HEADER(\$NAHE*INDEX\$))  ** + 1 \$  **********************************	BYTE(SPAGE MIDTH - 10.  5 \$) (HEADER(S&S)) = 5H(PAGE)  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(SNAME INDEXS))  * 1 \$  * TILE SN \$ ***FOR I = 0, 1, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, TITLE NO \$ **  ITLE SN \$ ***FOR I = 0, TITLE NO \$ **  ITLE SN \$ ***FOR	**************************************	**************************************	**************************************	BYTE(SPAGE MIDTH - 10.  5 \$) (HEADER(S&S)) = 5H(PAGE)  S *** **** **** **** **** **** **** *	**************************************
**************************************	*BYTE(SPAGE*MIDTH = 10.  * 5\$) (HEADER(\$6\$)) = 5H(PAGE)  * NAKE*BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME*INDEX\$))*  * 1 \$ *******************************	BYTE(SPAGE'WIDTH - 10.  5 \$) (HEADER(S63)) = 5H(PAGE)  8 NAKE'BYTE = MAKGIN +  1 LENGTH(HEADER(SNAME'INDEXS))  4 1 8  1 1 1 1 1 TILE'NO 8 "	ENTE(SPAGE MIDTH = 10.  5 \$) (HEADER(\$6.8.)) = 5H(PAGE).  5 **NAKE BYTE = MARGIN + 1.  **LENGTH(HEADER(\$NAME INDEX\$)).  **LENGTH(HEADER(\$NAME INDEX\$)].  **LENGTH(HEA	EBYTE(SPAGE MIDTH = 10.  5 \$) (HEADER(S&S)) = 5H(PAGE)  8 ***AME**BYTE = MAKGIN +  ** LENGTH(HEADER(SNAHE**INDEXS))  ****AME**BYTE = MAKGIN +  ***LENGTH(HEADER(SNAHE**INDEXS))  ****AME**BYTE = MAKGIN +  ***LENGTH(HEADER(SNAHE**INDEXS))  ****AME**BYTE = MAKGIN +  ***LENGTH(HEADER(SNAHE**INDEXS))  ***AME**BYTE = MAKGIN +  ***LENGTH(HEADER(SNAHE**INDEXS))  ***AME**BYTE = MAKGIN +  **AME**BYTE = MAKGIN +  **AME**BYT	ENTE(SPAGE MIDTH = 10.  5 \$) (HEADER(\$6\$)) = 5H(PAGE)  * NAHE BYTE = MARGIN +  * LENGTH (HEADER(\$NAHE * INDEX\$))  * 1 \$  ITLE*SN \$ **********************************	BYTE(SPAGE MIDTH = 10.  5 \$) (HEADER(\$6\$)) = 5H(PAGE)  8 \$  *NAHE BYTE = MAKGIN +  *LENGTH(HEADER(\$NAHE INDEX\$))  * 1 \$  **ILE*SN \$ **********************************	EBYTE(SPAGE MIDTH = 10.  5 \$) (HEADER(S&S)) = 5H(PAGE)  8 ***********************************	BYTE(SPAGE MIDTH - 10.  5 \$) (HEADER(S&S)) = 5H(PAGE)  8 *** *** *** *** *** *** *** *** ***	BYTE(SPAGE'WIDTH - 10.  5 \$) (HEADER(S63)) = 5H(PAGE)  *NAKE'BYTE = MAKGIN +  * LENGTH(HEADER(SNAME'INDEXS))  * 1 \$  **ILE'SN \$ ***********************************	**************************************	* SYTE(SPAGE MIDTH - 10.  * LENGTH (HEADER(SNAME INDEXS))  * LENGTH (HEADER(SNAME INDEXS)  * LENGTH (HEADER(SNAME INDEXS))  * LENGTH (HEADER(SNAME INDEXS)  * LENGTH (HEADER(SNAME INDEXS))  * LENGTH (HEADER(SNAME INDEXS)  * LENGTH	BYTE(SPAGE MIDTH = 10.  5 \$) (HEADER(\$6\$)) = 5H(PAGE)  * NAHE BYTE = MAKGIN +  * LENGTH (HEADER(\$NAHE * INDEX\$)) +  * 1 \$  * 1 \$  * 1 * 1 * 1 * 1 * 1 * 1 * 1 * 1 * 1 *	BYTE(SPAGE'WIDTH - 10.  5 \$) (HEADER(S63)) = 5H(PAGE)  *NAKE'BYTE = MAKGIN +  * LENGTH(HEADER(SNAME'INDEXS))  * 1 \$  *******************************	**************************************
**************************************	**************************************	**************************************	SAYTE (SPAGE MIDTH - 10.  5 \$) (HEADER (\$6.8.)) = 5H(PAGE )  * NAHE BYTE = MAKGIN +  * LENGTH (HEADER (\$NAHE INDEXS))  * 1 \$  ITLE SS \$ **FOR I = 0, 1, TITLE **NO \$ **FOR I = 0, 1, T	BYTE(SPAGE'WIOTH - 10.  5 \$) (HEADER(S63)) = 5H(PAGE).  8 **AME'BYTE = MAKGIN +  **LENGTH(HEADER(SNAME'INDEXS)).  ** 1 \$  **TLE'SN \$ ***********************************	BYTE(SPAGE MIDTH - 10.  5 \$) (HEADER(\$6\$)) = 5H(PAGE)  * NAHE BYTE = MAKGIN +  * LENGTH (HEADER(\$NAHE INDEX\$))  * 1 \$  ITLE*SN \$ **********************************	SYTE(SPAGE WIDTH - 10.  SS)(HEADER(SES)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH(HEADER(SNAME INDEXS))  + 1	### ### ##############################	### ### ##############################	**************************************	**************************************	SATE (SPAGE MIDTH - 10.  5 \$) (HEADER (\$6.8)) = 5H(PAGE)  *NAHE BATE = MARGIN +  *LENGTH (HEADER (SNAHE * INDEXS))  * 1 \$  ITLE SW \$ **********************************	BYTE(SPAGE WIDTH - 10.  5 \$) (HEADER(S63)) = 5H(PAGE)  8 ***AMÉ*********************************	### ### ##############################	**************************************
* BYTE (\$PAGE * MIDTH - 10.  * \$\$) (HEADER(\$6\$)) = 5H(PAGE)  * NAHE * BYTE = MAKGIN +  * LENGTH (HEADER(\$NAHE * INDEX\$))  * 1 \$	BYTE (SPAGE WIOTH - 10.  5 S) (HEADER(\$6\$)) = 5H(PAGE)  **NAKE BYTE = MAKGIN +  **LENGTH(HEADER(\$NAME INDEX\$))  **1 \$	**************************************	SYTE(SPAGE WIDTH - 10.  SS)(HEADER(SES)) = 5H(PAGE)  SSOCIATION - 10.  LENGTH(HEADER(SNAHE INDEXS))  1	### ### ##############################	BYTE(SPAGE WIDTH - 10.  SS)(HEADER(SES)) = 5H(PAGE)  SS)(HEADER(SNAHE INDEXS))  LENGTH HEADER(SNAHE INDEXS))  LAST SS S S S S S S S S S S S S S S S S S	### ### ##############################	### ### ##############################	HAYTE (SPAGE WIDTH - 10.  SS) (HEADER (S6S)) = 5H (PAGE)  NAKE BYTE = MAKGIN +  LENGTH (HEADER (SNAHE INDEXS))  + 1	**************************************	**************************************	### ### ##############################	### ### ##############################	**************************************	**************************************
**************************************	BYTE (SPAGE WIDTH - 10.  5 S) (HEADER(SGS)) = 5H(PAGE)  NAME BYTE = MARGIN +  LENGTH (HEADER(SNAME INDEXS))  1 S	BYTE(\$PAGE WIDTH - 10.  * SYTE(\$PAGE WIDTH - 10.  * SYNEE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAHE 'INDEX\$))  * 1	SYTE(SPAGE WIDTH - 10.  SS)(HEADER(SES)) = 5H(PAGE)  SS(HEADER(SNAME INDEXS))  LENGTH HEADER(SNAME INDEXS))  LA SS	### ### ##############################	SYTE(SPAGE WIDTH - 10.  SYTE(SPAGE WIDTH - 10.  SYNAKE BYTE = MAKGIN +  LENGTH(HEADER(SNAME INDEXS))  + 1	### ### ##############################	**************************************	**************************************	**************************************	**************************************	### ### ##############################	### ### ##############################	**************************************	**************************************
**************************************	BYTE (\$PAGE WIDTH - 10.  • 5\$) (HEADER(\$6\$)) = 5H(PAGE ).  * NAKE BYTE = MARGIN +  * LENGTH(HEADER(\$NAME INDEX\$)).  • 1 \$	**************************************	**************************************	**************************************	### ##################################	**************************************	**************************************	**************************************	**************************************	**************************************	ENTE(SPAGE WIDTH - 10.  SS)(HEADER(S6S)) = 5H(PAGE)  SS)(HEADER(SNAME INDEXS))  LENGTH(HEADER(SNAME INDEXS))	**************************************	**************************************	BYTE(SPAGE WIDTH - 10.  * 5\$) (HEADER(SGS)) = 5H(PAGE)  * NAME BYTE = MARGIN +  * LENGTH (HEADER(SNAME INDEXS))  * 1 \$  * 1 \$  * 1 \$  * 1 \$  * 1 * * * * * * * * * * * * * * * * *
**************************************	**************************************	**************************************	**************************************	*** SYTE (\$PAGE WIDTH - 10.  ** SYTE (\$PAGE WIDTH - 10.  *** SYTE (\$PAGE W	**************************************	*** SYTE (SPAGE WIDTH - 10.  *** SYTE (SPAGE WIDTH - 10.  *** SYTE (SAGE WIDTH - 10.	*** SYTE (\$PAGE *** NOT*** DIO*********************************	**************************************	**************************************	**************************************	**************************************	**************************************	**************************************	BYTE(\$PAGE WIDTH - 10.  * \$\$)(HEADER(\$6\$)) = 5H(PAGE).  * NAME BYTE = MARGIN +  * LENGTH(HEADER(\$NAME INDEX\$)).  * 1 \$  * 1 \$  * 1 \$  * 1 \$  * 1 * * * * * * * * * * * * * * * * *
**************************************	**************************************	**************************************	**************************************	***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  **	*** SYTE (SPAGE WIDTH - 10.  *** SYTE (SPAGE WIDTH - 10.  *** SYNAKE BYTE = MAKGIN +  *** LENGHHHEADER(SNAME INDEXS))  *** *** *** *** *** *** *** *** ***	***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  **	***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  ***  **  ***  ***  **	**************************************	**************************************	**************************************	BYTE (\$PAGE WIDTH - 10.  * \$\$)(HEADER(\$&\$)) = 5H(PAGE)*  * NAKE BYTE = MAKGIN +  * LENGTH (HEADER(\$NAME * INDEX\$))*  * 1 \$  ******************************	*** SYTE (SPAGE WIDTH - 10.  *** SYTE (SADER (SGS)) = 5H(PAGE)  *** NAKE BYTE = MARGIN +  *** LENGTH (HEADER (SNAHE 'INDEXS))  *** 1	**************************************	**************************************
BYTE (SPAGE MIDTH - 10.  5 \$) (HEADER(\$6\$)) = 5H(PAGE)  8 NAKE BYTE = HARGIN +  LENGTH(HEADER(\$NAME'INDEX\$))  4 + 1 \$  6	**************************************	**************************************	BYTE (SPAGE MIDTA - 10.  SS) (HEADER(\$6\$)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(\$NAME INDEX\$))  + 1 \$  LENGTH (HEADER(\$NAME INDEX\$))  + 1 \$  LENGTH (HEADER(\$NAME INDEX\$))  - 1 \$  LENGTH (HEADER(\$NAME INDEX\$))	**************************************	BYTE (SPAGE MIDTA - 10.  SS) (HEADER(\$6\$)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(\$NAME INDEX\$))  + 1  LENGTH (HEADER(\$NAME INDEX\$))  + 1  LENGTH (HEADER(\$NAME INDEX\$))  - 1  LENGTH (HEADER(\$NAME INDEX\$))	**************************************	**************************************	**************************************	BYTE (\$PAGE WIDTH - 10.  * \$\$)(HEADER(\$&\$)) = 5H(PAGE)*  * NAKE BYTE = MARGIN +  * LENGTH(HEADER(\$NAME INDEX\$))*  * 1 \$	BYTE (SPAGE MIDTH - 10.  * \$\$ (HEADER (\$6.3)) = 5H(PAGE )*  * NAKE BYTE = MAKGIN +  * LENGTH (HEADER (\$MAHE ' INDEX\$))*  * 1	BYTE (SPAGE MIDTA - 10.  SS) (HEADER(\$6\$)) = 5H(PAGE)  SNAHE BYTE = MAKGIN +  LENGTH (HEADER(\$NAME INDEXS))  + 1	### ### ##############################	BYTE (\$PAGE WIDTH - 10.  * \$\$)(HEADER(\$£)) = 5H(PAGE)*  * NAKE BYTE = MARGIN +  * LENGTH(HEADER(\$NAME INDEX\$))*  * 1 \$  * TLE SW \$ **********************************	ENTE(SPACE MIDTH = 10.  * \$\$ (HEADER(\$6\$)) = 5H(PAGE)  * NAME BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME INDEX\$))  * 1
* SYTE (\$PAGE* WIDTH = 10.  * SS) (HEADER(\$G\$)) = 5H(PAGE)*  * NAKE* BYTE = MAKGIN +  * LENGTH(HEADER(\$NAHE* INDEX*))*  * 1 \$	**************************************	BYTE (SPAGE MIDTA = 10,  SS (HEADER (SGS)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER (SNAME INDEXS))  1	**************************************	BYTE (SPAGE MIDTH - 10.  * \$\$ (HEADER(\$&\$1) = 5H(PAGE ).  * NAME BYTE = MARGIN +  * LENGTH (HEADER(\$MAME INDEX\$)).  * 1 \$  *******************************	BYTE (SPAGE WIDTH - 10.  SS) (HEADER(SGS)) = 5H(PAGE).  NAKE BYTE = MAKGIN +  LENGTH(HEADER(SNAME INDEXS)).  + 1	BYTE (SPAGE MIDTH - 10.  * \$\$ (HEADER(\$&\$1) = 5H(PAGE ).  * NAME BYTE = MARGIN +  * LENGTH (HEADER(\$MAME INDEX\$)).  * 1 \$ ********************************	ENTE (SPAGE MIDTH - 10.  * \$\$ (HEADER(\$&\$1) = 5H(PAGE)  * NAKE BYTE = MARGIN +  * LENGTH (HEADER(\$NAME INDEX\$))  * 1 \$  *******************************	ENTE(SPAGE MIDTH = 10,  \$ \$ (HEADER(\$6\$)) = 5H(PAGE)  **NAME BYTE = MARGIN +  **LENGTH(HEADER(\$NAME INDEX\$))  ***ILE SW \$ **********************************	BYTE (SPAGE MIDTA - 10.  SS (HEADER (SGS)) = SH(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER (SNAME INDEXS))  LENGTH (HEADER (SNAME INDEXS))  LILE'SM SFOR I = 0, 1, TITLE'NO S  ITLE'SM S	BYTE (SPAGE MIDTA - 10.  SS) (HEADER (SGS)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER (SNAME INDEXS))  + 1	BYTE (SPAGE WIDTH - 10.  SS) (HEADER(SGS)) = 5H(PAGE)  NAKE BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))  + 1	**************************************	BYTE (SPAGE MIDTA - 10.  \$ \$ (HEADER (\$6.8)) = 5H(PAGE).  **NAME BYTE = MAKGIN +  **LENGTH (HEADER (\$NAME * INDEX\$)).  **1 \$  **ILE*SM \$ ***FOR I = 0. 1, TITLE*NO \$ **	BYTE (SPAGE WIDTH - 10.  SS) (HEADER(\$6\$)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(\$NAME INDEXS))  + 1
**************************************	**************************************	BYTE (SPAGE WIDTH - 10.  S\$) (HEADER (\$6.8)) = 5H(PAGE)  **NAHE BYTE = MAKGIN +  **LENGTH (HEADER (\$MAHE * INDEX\$))  **1.**  ITLE SW \$ **********************************	**************************************	SATE (SPAGE WIDTH - 10.  SS) (HEADER(SS)) = 5H(PAGE).  SAME BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS)).  1	**************************************	SAYTE (SPAGE WIDTH - 10.  SS) (HEADER(SS)) = 5H(PAGE)  NAKE BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))  1 I = 0. 1. OUT BUF SIZE - 1.  THE SA SFOR I = 0. 1. TITLE NO S  THE SA SFOR I = 0. TITLE NO S  THE SA SFOR I = 0. TITLE NO S  THE SA SFOR I = 0. TITLE NO S  THE SA SFOR I = 0. TITLE NO S  THE SA SFOR I = 0. TITLE NO S  THE SA SFOR I = 0. TITLE NO S  THE SA SFOR I = 0. TITLE NO S  THE SA SFOR I = 0. TITLE NO S  THE SA SFOR I = 0. TITLE NO S  THE SA SFOR I = 0. TITLE NO S  THE S	BYTE (SPAGE MIDTA - 10.  SS) (HEADER (SGS)) = SH(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER (SNAME INDEXS))  1	BYTE (SPAGE WIDTH - 10.  \$ 5\$) (HEADER(\$6\$)) = 5H(PAGE)  **NAHE BYTE = MAKGIN +  **LENGTH (HEADER(\$NAHE * INDEX\$))  **1  ITLE SM \$ ***FOR I = 0, 1, TITLE **NO \$ **  ITLE **ITLE **NO **  I = 0, 1, OUT **BUF **SIZE - 1 ***  **ITLE	BYTE (SPAGE WIDTH - 10.  SS) (HEADER(SGS)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))  + 1	SYTE (SPAGE WIDTH - 10.  SS) (HEADER(SS)) = 5H(PAGE)  NAKE BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))  TILE SM SFOR I = 0. 1. TITLE NO S  ITLE SM SFOR I = 0. 1. TITLE NO S  ITLE SM SFOR I = 0. 1. TITLE NO S  I = 0. 1. OUT BUF SIZE - 1.  "FOUT LINE (SIZE - 1.)  "FOUT LINE (SIZE - 1.)	SECTION STATE STAT	SAYTE (SPAGE WIDTH - 10.  SS) (HEADER(SGS)) = 5H(PAGE)  NAKE BYTE = MAKGIN +  LENGTH (HEADER(SNAME 'INDEXS))  1 I = 0. 1. OUT BUF 'SIZE - 1.  THE 'SY SFOR I = 0. 1. TITLE' NO S  THE 'SY S	BYTE (SPAGE WIDTH - 10.  SS) (HEADER (SGS)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER (SNAME INDEXS))  TILE SM SFOR I = 0. 1. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFOR I = 0. TITLE NO S  ITLE SM SFO	### ##################################
**************************************	*** STORY I = U. I. HEAD'NO S.  *** SYTE (SPAGE WIDTH - 10.  *** SS) (HEADER(\$CS)) = 5H(PAGE)**  *** LENGTH(HEADER(\$NAME'INDEXS))**  *** LENGTH(HEADER(\$NAME'INDEXS))**  *** LENGTH(HEADER(\$NAME'INDEXS))**  *** STORY I = U. I. TITLE'NO S.**	**************************************	**************************************	THE STATE ST	**************************************	TILE'SM \$FOR I = 0, 1, TILE'NO \$FOR I =	TILE'SW SFOR I = 0, 1, TILE'NO S  ITLE'SW SFOR I = 0, 1, TILE'SW S  ITLE'SW S	SECOND SE	*** SYTE (\$PAGE *** IDTA - 10.*** *** SY	**************************************	**************************************	TILE SW SFOR I = 0, 1, TILE NO SFOR I =	### STOR I = U : I HEADT NO S  ### SETTE ### STE	*** SYTE (\$PAGE *** IDTA - 10.
**************************************	*** FOR I = 0. 1. HEAD'NO \$  ***********************************	POR I = 0, 1, HEAD'NO S  ***********************************	FOR I = 0. I. HEAD'NO S FINE SS (46 D) = 54 (PAGE) F SS (46 D) E SH (PAGE) F SS (46 D) E S	FOR I = 0. I. HEAD'NO S  *******************************	TALE SW & **********************************	FOR I = 0. I. HEAD'NO S FINE STATE S	FOR I = 0. I. HEAD'NO S FINE STATE S	FOR I = 0. I. HEAD'NO S FINE STATE S	*** FOR I = 0. I. HEAD'NO S *****  *** SYTE (SPAGE *** IOTA *** 10.***  *** SYTE (SPAGE *** IOTA *****  *** SYTE (SPAGE ***********  *** SYTE (SPAGE ************************************	**************************************	**************************************	FOR I = 0. 1. HEAD'NO S FINE SS (SE MIDIA - 10. SS (HEADER(SES)) = 5H(PAGE) SS (HEADER(SES)) = 5H(PAGE) SS (HEADER(SES)) = 5H(PAGE) SS (HEADER(SES)) = 5H(PAGE) SS (HEADER(SES)) = 1. SS (HEADER(SES)) SS (HEADER(	POR I = 0. I. HEAD'NO S  BYTE(SPACE'MIDTM - 10.  S (HEADER(SGS)) = 5H(PACE)  NAME BYTE = MAKGIN +  LENGTH(HEADER(SNAME'INDEXS))  + 1 S  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	**************************************
**************************************	**************************************	**************************************	FOR I = 0, 1, HCAD*NO \$ *  *******************************	FOR I = 0, 1, HCAD"NO \$ "  "BYTE(SPACE"MIDTH - 10,  "S\$ (HCADER(S&S)) = 5H(PACE)  "NAKE BYTE = MAKGIN +  "LENGTH(HEADER(SNAME"INDEXS));  "+ 1	FOR I = 0, 1, HCAD"NO \$ "  *********************************	FOR I = 0, 1, HCAD*NO \$ *****  *****************************	FOR I = 0, 1, HCAD"NO \$ "  "BYTE (SPAGE MIDTH - 10,  "SA (HCADER(S&1)) = 5H(PAGE)  "NAKE BYTE = MAKGIN +  "LENGTHHEADER(SNAME INDEXS))  " 1 \$  "TLE"SN \$ "FOR I = 0, 1, TITLE"NO \$ "  "TTLE"SN \$ "FOR I = 0, 1, TITLE"NO \$ "  "TTLE"NO \$ "FOR I = 0, 1, TITLE"NO \$ "  "TTLE"SN \$ "FOR I = 0, 1, TITLE"SN \$ "	FOR I = 0, 1, HCAD"NO \$ "  "STEER STEER	FOR I = 0, 1, HCAD*NO \$ *****  *****************************	FOR I = 0, 1, HCAD*NO \$ *****  *****************************	FOR I = 0, 1, HCAD*NO \$ ***  ******************************	FOR I = 0, 1, HCAD*NO \$ *****  *****************************	FOR I = 0, 1, HCAD*NO \$ *****  *****************************	FOR I = 0.1, HCAD*NO \$ *  *******************************
### ### ##############################	**************************************	**************************************	**************************************	FOR I = 0, 1, HEAD*NO & "  ********************************	FOR I = 0, 1, HEAD*NO & "  ********************************	FOR I = 0, 1, HEAD NO S "  *********************************	FOR I = 0, 1, HEAD*NO & "  ********************************	FOR I = 0, 1, HEAD*NO \$ "  *********************************	FOR I = 0, 1, HEAD*NO \$ "  ********************************	**************************************	### ### ##############################	**************************************	FOR I = 0, 1, HEAD*NO \$ "  *********************************	**************************************
PFOR I = 0, 1, HEAD'NO S ****  ******************************	*** SET TO THE T	**************************************	**************************************	**************************************	**************************************	**************************************	**************************************	**************************************	**************************************	**************************************	### ### ##############################	**************************************	**************************************	### ### ##############################
PFOR I = 0, 1, HEAD'NO S "  " SYLE (SPACE'N IDIM - 10,  " SS) (HEADER(\$6\$)) = 5H(PAGE)  " LENGTH (HEADER(\$NAME'INDEX\$))  " + 1 S  " **********************************	*** SECONT TO THE TO THE STATE OF THE STATE	**************************************	""  ""  ""  ""  ""  ""  ""  ""  ""  ""	**************************************	**************************************	**************************************	PFOR I = 0, 1, HEAD************************************	*** **********************************	*** **********************************	*** SYTE (\$PAGE WIDTH - 10.  *** SYTE (\$PAGE WIDTH - 10.  *** SYSTE (\$PATE = MAKGIN + 10.  *** LENGTH (HEADER(\$MAME ** INDEX*))	### ##################################	**************************************	*** **********************************	### ### ##############################
**************************************	**************************************	**************************************	**************************************	**************************************	*** *** *** *** *** *** *** *** *** **	**************************************	**************************************	**************************************	**************************************	**************************************	PFOR I = 0, 1, HEAD"NO \$ "  BYTE (SPAGE MIDTM - 10,  S S) (HEADER(SES)) = SH(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))  TILESM \$ "	*** *** *** *** *** *** *** *** *** **	**************************************	BYTE (SPAGE MIDTH - 10.  * \$\$) (HEADER(\$&\$)) = 5H(PAGE)  * NAKE BYTE = MAKGIN +  * LENGTH (HEADER(\$NAME * INDEX\$))  * 1
**************************************	*** FOR I = 0, 1, HEAD************************************	BYTE (SPAGE WIDTH - 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,	PFOR I = 0, 1, HEAD NO S PROFESSION S PROFES	ENTE(EPAGE WIDTH - 10.  * \$\$ (HEADER(\$&\$)) = 5H(PAGE)  * * ********************************	PFOR I = 0, 1, HEAD"NO \$ ""  ***SYTE(\$PAGE"WIDTH - 10, **  ***S	FOR I = 0, 1, HCAD************************************	ENTE(EPAGE WIDTH - 10.  * \$\$ (HEADER(\$&\$)) = 5H(PAGE)  * NAME BYTE = MARGIN +  * LENGTH(HEADER(\$MAME INDEX\$))  * 1 \$  *******************************	ENTE(EPAGE MIDTH - 10, 1, HEAD" NO S ""  **SANTE(SPAGE MIDTH - 10, 1, SANTE BYTE = MAKGIN + LENGTH(HEADER(SNAME INDEXS))  **LENGTH(HEADER(SNAME INDEXS)  **LENGTH(HEADER(SNAME INDEXS))  **LENGTH(HEADER(SNAME INDEXS)  **LENGTH(HEADER(SNAME INDEXS)  **LENGTH(HEADER(SNAME INDEXS)  **LENGTH(HEADER(SNAME INDEXS)  **LENG	FOR I = 0, 1, HEAD" NO \$ "  BYTE (\$PAGE MIDTM - 10,  \$ \$ (HEADER(\$6\$)) = 5H(PAGE)  "NAME BYTE = MAKGIN +  "LENGTH (HEADER(\$NAME "INDEX\$))  " 1 \$ \$ (HENGTH (HEADER(\$NAME "INDEX\$))  " 1 \$ \$ (HENGTH (HEADER(\$NAME "INDEX\$))  " 1 \$ (HENGTH (HEADER(\$NAME	PFOR I = 0, 1, HEAD" NO S ""  ********************************	PFOR I = 0, 1, HEAD NO S PROPERTY OF STATE OF ST	FOR I = 0, 1, HEAD NO S  *******************************	FOR I = 0, 1, HEAD NO S  BYTE (SPAGE MIDTA - 10,  SS (HEADER(SES)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))	FOR I = 0, 1, HEAD NO S  BYTE (SPAGE WIDTH - 10,  SS) (HEADER(SES)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))  TILE SM SFOR I = 0, 1, TITLE NO S  ITLE SM SFOR I = 0, 1, TITLE NO S  I = 0, 1, OUT BUF SIZE - 1 * ********************************
## ## ## ## ## ## ## ## ## ## ## ## ##	BYTE (SPAGE WIDTH - 10.  Syl (HEADER(\$6\$)) = 5H(PAGE)  NAKE BYTE = MAKGIN +  LENGTH(HEADER(\$NAME INDEXS))  + 1 \$  THE STREET OF 11 TILE NO S SEED	### ### ##############################	**************************************	FOR I = 0, 1, HEAD'NO \$ "	### ### ##############################	FOR I = 0, 1, HEAD'NO \$ "	EFOR I = 0, 1, HEAD'NO S PRICE SPACE MIDTH - 10, 5 S) (HEADER(\$6\$)) = 5H(PACE)  **NAME BYTE = MAKGIN + LENGTH(HEADER(\$NAME INDEX\$))  **LENGTH(HEADER(\$NAME INDEX\$))  **LENGTH(HEADER(\$NAME INDEX\$))  **ILLE SW & **********************************	### ### ##############################	### ### ##############################	BYTE (SPAGE WIDTH - 10.  **SYTE (SPAGE WIDTH - 10.  **SAS (HEADER(\$6.8)) = 5H(PAGE)  **NAKE BYTE = MAKGIN +  **LENGTH(HEADER(\$NAME INDEXS))  **********************************	BYTE (SPAGE WIDTH - 10.  SYTE (SPAGE WIDTH - 10.  SYN (HEADER(SEN)) = 5H(PAGE)  NAKE BYTE = MAKGIN +  LENGTH(HEADER(SNAME INDEXS))  1	### ### ##############################	### ### ##############################	### ### ##############################
FOOR I = 0, 1, HEAD'NO S F  FOOR I = 0, 1, HEAD'NO S  FOOR I = 0, 1, HEAD'NO S  FOOR I = 0, 1, HEAD'NO S  FOO I = 0, HE	FOR I = 0, 1, HZAD'NO \$ "  *** *** *** *** *** *** *** *** ***	FOR I = 0, 1, HcAD*NO & *  *******************************	FOR I = 0, 1, HEAD'NO \$ "  *********************************	FOR I = 0.1, HEAD'NO \$ "  *********************************	FOR I = 0, 1, HEAD'NO \$ "  *********************************	FOR I = 0.1, HEAD'NO \$ "  *********************************	FOR I = 0.1, HEAD'NO \$ "  *********************************	FOR I = 0, 1, HcAD*NO & **********************************	FOR I = 0, 1, HEAD'NO \$ "  *********************************	FOR I = 0, 1, HEAD'NO \$  ********************************	FOR I = 0, 1, HcAD"NO \$ "  *********************************	FOR I = 0, 1, HEAD'NO & "  *********************************	FOR I = 0.1, HZAD'NO \$  *******************************	FOR I = 0.1, HEAD'NO \$  *******************************
# FOR I = 0. 1, HEAD'NO \$ "  ********************************	#FOR I = 0, 1, HEAD'NO \$ "  ***SYTE (\$PAGE WIDTH - 10,  ***S\$) (HEADER(\$G\$)) = 5H(PAGE )  ****LENGTH(HEADER(\$NAME'INDEX\$))  ****LENGTH(HEADER(\$NAME'INDEX\$))  ****LENGTH(HEADER(\$NAME'INDEX\$))  ****LENGTH(HEADER(\$NAME'INDEX\$))  ****LENGTH(HEADER(\$NAME'INDEX\$))	FOR I = 0, 1, HEAD'NO S "  *********************************	FOR I = 0, 1, HEAD'NO & "  *********************************	FOR I = 0, 1, HEAD'NO & "  "STATE (SPAGE "MIDTH - 10,  "SA) (HEADER(SAS)) = 5H(PAGE)  "NAKE BYTE = MAKGIN +  "LENGTHHEADER(SNAME INDEXS))  "TLE"SN & "FOR I = 0, 1, TITLE"NO & "  "TTTLE"SN & "FOR I = 0, 1, TITLE"SN & "  "TTTLE"SN & "FOR I = 0, 1, TITLE"SN & "  "TTTLE"SN & "FOR I = 0, 1, TITLE"SN & "  "TTTLE"SN & "FOR I = 0, 1, TITLE"SN & "  "TTTLE"SN & "FOR I = 0, 1, TITLE"SN & "  "TTTLE"SN & "FOR I = 0, 1, TITLE"SN & "  "TTTLE"SN & "FOR I = 0, 1, TITLE"SN & "  "TTTLE"SN & "FOR I = 0, 1, TITLE"SN & "  "TTTLE"SN & "FOR I = 0, 1, TITLE"SN & "  "TTTLE"SN & "FOR I = 0, 1, TITLE"SN & "  "TTTLE"SN & "FOR I = 0, 1, TITLE"SN & "  "TTTLE"SN & "FOR I = 0, 1, TITLE"SN & "  "TTTLE"SN & "FOR I = 0, 1, TITLE"SN & "  "TTTLE"SN & "FOR I = 0, 1, TITLE"SN & "  "TTTLE"SN & "FOR I = 0, 1, TITLE"SN & "	FOR I = 0, 1, HEAD'NO & "  *********************************	FOR I = 0, 1, HEAD'NO & "  "BYTE (SPAGE "MIDTH - 10,  "SYNAKE BYTE = MAKGIN +  "LENGTHHEADER(SNAME INDEXS))  "A 1	FOR I = 0, 1, HEAD'NO & "  *********************************	FOR I = 0, 1, HEAD'NO S "  *********************************	FOR I = 0, 1, HEAD'NO S "  "BYTE (SPAGE "MIDTH - 10,  "S NAKE BYTE = MAKGIN +  " LENGTH HEADER (SNAME "INDEXS))  " 1	FOR I = 0, 1, HEAD'NO \$ "  *********************************	FOR I = 0. 1, HEAD'NO S "  *********************************	FOR I = 0, 1, HEAD'NO & "  " FOR I = 0, 1, HEAD'NO & "  " BYTE (SPAGE "MIDTH - 10, 1  " SA) (HEADER(SA)) = 5H(PAGE)  " SA) (HEADER(SA)) = 5H(PAGE) " SA) (HEADER(SA)) = 5H(PAGE) " S	FOR I = 0, 1, HEAD'NO S "  *********************************	## ## ## ## ## ## ## ## ## ## ## ## ##
FOR I = 0, 1, HEAD'NO \$ "  BYTE (\$PAGE'NIDTH - 10,  S\$) (HEADER(\$G\$)) = 5H(PAGE)  NAKE BYTE = HARGIN +  LENGTH (HEADER(\$NAME'INDEX\$))  + 1 \$  **********************************	FOR I = 0, 1, HCAD"NO \$ "  *********************************	** FOR I = 0, 1, HEAD**NO \$ *****  *****************************	FOR I = 0, 1, HEAD*NO \$ *****  *****************************	FOR I = 0, 1, HEAD'NO \$ ****  ******************************	FOR I = 0, 1, HEAD*NO \$ *****  *****************************	FOR I = 0, 1, HEAD'NO S  ***SYTE (SPACE WIDTH - 10,  ***SYTE (SPACE WIDTH - 10,  ***SYTE (SPACE WIDTH - 10,  ***AKE BYTE = MAKGIN +  ***LENGHHHEADER(SNAME'INDEXS))  **********************************	FOR I = 0, 1, HEAD'NO \$ ****  ******************************	** FOR I = 0, 1, HEAD**NO \$ *****  ** **************************	** FOR I = 0, 1, HEAD**NO \$ *********************************	FOR I = 0, 1, HEAD'NO \$ "  BYTE (\$PAGE WIDTH - 10,  \$ \$ (HEADER(\$E)) = 5H(PAGE)  **NAKE BYTE = MARGIN +  **LENGTH(HEADER(\$NAME'INDEX\$))  **1 \$  **LENGTH(HEADER(\$NAME'INDEX\$))  **1 \$	FOR I = 0, 1, HEAD*NO \$ ****  ******************************	FOR I = 0, 1, HEAD'NO S  ***********************************	**FOR I = 0, 1, HEAD**NO \$ *********************************	FOR I = 0, 1, HEAD"NO \$ "  BYTE (\$PAGE WIDTH - 10,  \$ \$ (HEADER(\$&\$)) = 5H(PAGE )  "NAKE BYTE = MAKGIN +  "LENGTH (HEADER(\$NAME "INDEX\$))  " LENGTH (HEADER(\$NAME "INDEX\$))  " LENGTH (HEADER(\$NAME "INDEX\$))  " LENGTH (HEADER(\$NAME "INDEX\$))  " LENGTH (HEADER(\$NAME "INDEX\$))  " ITLE SW \$ "FOR I = 0, 1, TITLE "NO \$ "  " ITLE SW \$ "FOR I = 0, 1, TITLE "NO
FOR I = 0, 1, HEAD'NO \$ "  BYTE (\$PAGE'NIDTH - 10,  S\$) (HEADER(\$G\$)) = 5H(PAGE)  NAKE BYTE = MARGIN +  LENGTH(HEADER(\$NAME'INDEX\$))  + 1 \$	FOR I = 0, 1, HCAD'NO \$ ****  ******************************	FOR I = 0, 1, HEAD"NO \$ "  *** BYTE (\$ PAGE "M DIM - 10,  *** S\$ (HEADER(\$ L\$) = 5H(PAGE )  *** S** CHOOLE STEEN STEE	FOR I = 0, 1, HCAD*NO \$ *****  *****************************	FOR I = 0, 1, HEAD'NO \$  *** *** *** *** *** *** *** *** *	FOR I = 0, 1, HCAD"NO \$ "  *** SYTE (SPACE WIDTH - 10,  *** SYTE (SPACE WIDTH - 10,  *** SYNAKE BYTE = MAKGIN +  *** LENGTH (HEADER(SNAME INDEXS))  *** *** *** *** *** *** *** *** ***	FOR I = 0, 1, HEAD'NO \$  ***SYTE (SAGE WIDTH - 10,  **SYTE (SAGE WIDTH - 10,  ***SYTE (SAGE WIDTH	FOR I = 0, 1, HEAD'NO \$  *******************************	FOR I = 0, 1, HEAD"NO \$ "  BYTE (\$AAE" WIDTH - 10,  \$ \$ (HEADER(\$L\$)) = 5H(PAGE )  **NAKE BYTE = MAKGIN +  **LENGTH(HEADER(\$NAME"INDEX\$))	FOR I = 0, 1, HEAD'NO \$ "  BYTE (\$PAGE WIDIM - 10,  \$ \$ (HEADER(\$E)) = 5H(PAGE)  **NAKE BYTE = MARGIN +  **LENGTH(HEADER(\$NAME'INDEX\$))  **1 \$  **LENGTH(HEADER(\$NAME'INDEX\$))  **1 \$	FOR I = 0, 1, HEAD'NO \$ "  BYTE (EPAGE MIDTH - 10,  BYTE (EPAGE MIDTH - 10,  SANAKE BYTE = MAKGIN +  LENGTH (HEADER (SANAKE NOEXS))  LENGTH (SANAKE NOEXS)  LENGTH (SANAKE N	FOR I = 0, 1, HcAD'NO \$ "  BYTE (SPAGE MIDTH - 10,  SS) (HcADER(SAS)) = SH(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HcADER(SNAME'INDEXS))  * LENGTH (Hcader)  * Length (Hcad	FOR I = 0, 1, HEAD'NO \$  ***SYTE (SPACE'NIDIA - 10,  ***SYTE (SPACE'NIDIA - 10,  ***SYTE (SPACE'NIDIA - 10,  ***SYTE (SPACE'NIDEX*))  ***AKE'BYTE = MAKGIN +  ***LENGTH(HEADER(SNAHE'INDEX*))  *********************************	FOR I = 0, 1, HEAD"NO \$ "  BYTE (\$PAGE WIDTH - 10,  \$ \$ (HEADER(\$&B)) = 5H(PAGE)  **NAKE BYTE = MARGIN +  **LENGTH(HEADER(\$MAME"INDEX\$))  **1 \$ ********************************	FOR I = 0, 1, HEAD'NO \$ "  BYTE (EPAGE MIDTH - 10,  S\$ (HEADER(S&3)) = 5H(PAGE)  NAKE BYTE = MAKGIN +  LENGTH (HEADER(SNAME' INDEXS))  LENGTH (HEADER(SNAME
**************************************	FOR I = 0, 1, HCAD*NO \$ **  ******************************	FOR I = 0, 1, HCAD NO S  ***SYTE (\$FAGE ***IDTH - 10,	FOR I = 0, 1, HEAD'NO S ""  ********************************	FOR I = 0, 1, HCAD****  *******************************	FOR I = 0, 1, HEAD'NO S "  ***SYTE(SPAGE'WIOTH - 10, **  ***SYTE(SPAGE'WIOTH - 10, **  ***SYTE(SPAGE'WIOTH - 10, **  ***SYTE = MAKGIN + **  ***LENGTH(HEADER(SNAME'INDEXS))*  *********************************	FOR I = 0, 1, HCAD'NO \$  *******************************	FOR I = 0, 1, HCAD****  *******************************	FOR I = 0, 1, HCAD****  ***SYTE(SPAGE************************************	FOR I = 0, 1, HCAD****  ***SYTE(SPAGE***IDTM**-10, ***  ***S*****************************	FOR I = 0, 1, HCAD************************************	FOR I = 0, 1, HCAD****  *******************************	FOR I = 0, 1, HCAD'NO S *  *******************************	FOR I = 0.1, HCAD****  ***SYTE(SPAGE***MIDTH***-10.**  ***S******************************	FOR I = 0, 1, HEAD NO S  BYTE (SPAGE MIDTH - 10,  SS) (HEADER(\$6\$)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(\$NAME INDEXS))  + 1 S  TILE SN SFOR I = 0, 1, TITLE NO S  I = 0, 1, OUT BUF SIZE - 1 **********************************
**************************************	**************************************	FOR I = 0, 1, HEAD NO & "	**************************************	FOR I = 0, 1, HEAD NO \$ "  *********************************	FOR I = 0, 1, HEAD'NO S "	FOR I = 0, 1, HEAD NO S "  *********************************	FOR I = 0, 1, HEAD NO \$ "  BYTE (EPAGE MIDTH = 10,  SA) (HEADER(SA) = 5H(PAGE)  NAKE BYTE = MARGIN +  LENGTH (HEADER(SNAME INDEXS))	FOR I = 0, 1, HEAD NO & "  BATE (SPAGE MIDTA - 10,  SS) (HEADER(SES)) = SH(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))  LILE'SN & "	FOR I = 0, 1, HEAD NO & "  BYTE (SPAGE MIDTH - 10,  SS) (HEADER(\$6\$)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(\$NAME * INDEXS))  * 1	FFOR I = 0, 1, HEAD'NO S ""  ********************************	FOR I = 0, 1, HEAD'NO \$ "	FOR I = 0, 1, HEAD'NO S ""  *******************************	FOR I = 0, 1, HEAD NO & "  BYTE (SPAGE MIDTM - 10,  S\$) (HEADER(\$6\$)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(\$NAME INDEX\$))	FOR I = 0, 1, HEAD'NO \$ "
FOR I = 0, 1, HEAD'NO S ****  ******************************	FOR I = 0, 1, HCAD"NO S PROFESSION S PROFESS	FOR I = 0.1. HEAD'NO S BYTE (SPACE'NIOTH - 10.  SYTE (SPACE'NIOTH - 10.  SS) (HEADER(SES)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(SNAME'INDEXS))  LENGTH (HEADER(SNAME'INDEXS))  LENGTH (HEADER(SNAME'INDEXS))  LENGTH (HEADER(SNAME'INDEXS))  LENGTH (HEADER(SNAME'INDEXS))  LETTE'SN S "	FOR I = 0, 1, HEAD'NO S ****  ******************************	FOR I = 0, 1, HEAD'NO S *  *******************************	FOR I = 0, 1, HEAD'NO S ****  ******************************	FOR I = 0, 1, HEAD'NO S * ********************************	FOR I = 0, 1, HEAD'NO S *  *******************************	FOR I = 0.1, HEAD'NO S PROPERTY OF STATE SPACE WIDTH - 10.  SYTE (SPACE WIDTH - 10.  SANKE BYTE = MAKGIN +  LENGTH (HEADER (SANAHE' INDEXS))  * LENGTH (HEADER (SNAHE' INDEXS))	FOR I = 0.1, HEAD'NO S B  *********************************	FOR I = 0, 1, HEAD'NO S PROPERTY OF THE STATE ST	FOR I = 0, 1, HEAD'NO S ****  ******************************	FOR I = 0, 1, HEAD'NO S * ********************************	FOR I = 0.1, HEAD'NO S PROPERTY OF STATE SPACE WIDTH - 10.  SS)(HEADER(SGS)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH(HEADER(SNAME INDEXS))  + 1	FOR I = 0, 1, HEAD'NO S PROPERTY OF STATE SPACE WIDTH - 10, S PHIPAGE SPACE WIDTH - 10, S PHIPAGE SPACE WIDTH - 10, S PHIPAGE SPACE
*FOR I = 0.1, HEAD'NO \$ * ********************************	FOR I = 0. 1, HEAD'NO \$ ****  ******************************	FOR I = 0, 1, HCADONO S *  *** *** *** *** *** *** *** *** *	FOR I = 0.1, HEAD'NO \$ "  *********************************	FOR I = 0.1, HcAD'NO \$ "  "FOR I = 0.1, HcAD'NO \$ "  "SYIE (EPAGE MIDTH - 10.  "LENGTH (HEADER (SNAHE INDEXS))  "LENGTH (HEADER (SNAHE INDEXS))  "TILE SY \$ "FOR I = 0.1, TITLE NO \$ "  "TILE SY \$ "FOR I = 0.1, TITLE NO \$ "  "ITLE SY \$ "-	FOR I = 0.1, HEAD'NO \$ "  *** *** *** *** *** *** *** *** ***	FOR I = 0.1, HcAD'NO \$ "  *********************************	FOR I = 0.1, HcAD*NO \$ *  *******************************	FOR I = 0. 1, HCAD*NO \$ *  *******************************	FOR I = 0.1, HEAD'NO \$ "  *********************************	FOR I = 0. 1, HEAD'NO \$ "  *********************************	FOR I = 0.1, HEAD'NO \$ "  *********************************	FOR I = 0.1, HEAD'NO \$ *  *******************************	FOR I = 0.1, HEAD'NO \$  *** *** *** *** *** *** *** *** **	FOR I = 0. 1. HEAD'NO \$ "  *********************************
FOR I = 0.1, HEAD'NO \$ ""  ********************************	#FOR I = 0. 1, HEAD'NO \$ "  ***BYTE (\$PAGE WIDTH - 10.  ***S\$) (HEADER(\$G\$)) = 5H(PAGE )  ****AKE BYTE = MAKGIN +  ***LENGTH(HEADER(\$NAME INDEX\$))  ****INDEX***  ****INDEX**  ***	FOR I = 0, 1, HEAD'NO S " FOR I = 0, 1, HEAD'NO S " FOR I = 0, 1, HEAD'NO S " FOR I = 0, 1, TITLE'NO S "	FOR I = 0.1, HcAD*NO \$ *  *******************************	FOR I = 0.1, HEAD'NO \$ "  "FOR I = 0.1, HEAD'NO \$ "  "BYTE (SPAGE MIDTH - 10.  "SANAKE BYTE = MAKGIN +  "LENGTH (HEADER (SNAME INDEXS))  "A 1 \$  "TLE SN \$ "FOR I = 0.1, TITLE NO \$ "  "TLE SN \$ "FOR I = 0.1, TITLE NO \$ "  "THE SN \$ "FOR I = 0.1, TITLE NO \$ "	FOR I = 0.1, HCAD*NO \$  ********************************	FOR I = 0.1, HEAD'NO \$ "  *********************************	FOR I = 0. 1, HEAD'NO S " FOR I = 0. 1, HEAD'NO S " BYTE (SPAGE'MIDTH - 10.  S S (HEADER(SAS)) = 5H(PAGE)  S S (HEADER(SANME'INDEXS))  * A L S S (HEADER(SNAME'INDEXS))  * 1 L S S S S S S S S S S S S S S S S S S	FOR I = 0, 1, HEAD'NO S " "STEER STEER	FOR I = 0, 1, HEAD'NO S "  *********************************	FOR I = 0. 1, HCAD*NO \$  *******************************	FOR I = 0.1, HCAD************************************	FOR I = 0.1, HEAD'NO S "  *********************************	FOR I = 0, 1, HEAD'NO S "  *********************************	FOR I = 0.1, HCAD*NO \$  ********************************
# FOR I = 0, 1, HEAD*NO \$ *****  *****************************	FOR I = 0, 1, HEAD"NO \$ ""  BYTE (\$PAGE WIDIM - 10,  \$ \$ (HEADER(\$&\$)) = 5H(PAGE )  "NAKE BYTE = MARGIN +  "LENGTH(HEADER(\$NAME INDEX\$))  " 1 \$  "	PFOR I = 0, 1, HEAD*NO \$ ***  ***RYTE(\$PAGE***HDT#** - 10,	FOR I = 0, 1, HEAD'NO S "  *** SETTE (SPAGE MIDTH - 10, 1)  *** SETTE (SPAGE MIDTH - 10, 1)  *** SETTE (SPAGE MIDTH - 10, 1)  *** LENGTH (HEADER(SNAHE INDEXS))  *** LENGTH (HEADER(SNAHE INDEXS))  *** LENGTH (HEADER(SNAHE INDEXS))  *** LENGTH (HEADER(SNAHE INDEXS))  *** LENGTH (LENGTH INDEXS)  *** LENGTH (LENGTH INDEXS)  *** LENGTH (LENGTH INDEXS))  *** LENGTH (LENGTH INDEXS)  *** LENGTH (LENGTH INDEXS)  *** LENGTH (LENGTH INDEXS)  *** LENGTH (LENGTH INDEXS)  *** LENGTH (LE	FOR I = 0, 1, HEAD'NO S ***********************************	FOR I = 0, 1, HEAD'NO S ""  ***SYTE(SPAGE'MIDTH - 10, S STANDERS) = 5H(PAGE)  ***SANAHE'BYTE = MAKGIN + LENGTH'HEADER(SNAHE'INDEXS))  ****LENGTH'HEADER(SNAHE'INDEXS))  ****LENGTH'HEADER(SNAHE'INDEXS)  ****LENGTH'HEADER(SNAHE'INDEXS)  ****LENGTH'HEADER(SNAHE'INDEXS)  ****LENGTH'HEADER(SNAHE'INDEXS)  ****LENGTH'HEADER(SNAHE'INDEXS)  ****LENGTH'HEADER(SNAHE'INDEXS)  ****LENGTH'	FOR I = 0, 1, HEAD'NO S ***********************************	FOR I = 0, 1, HEAD'NO S ***********************************	FOR I = 0, 1, HEAD*NO S *****  *****************************	FOR I = 0, 1, HEAD'NO S * *** SYTE (\$PAGE *** NOT** - 10, *** *** SYTE (\$PAGE *** NOT** - 10, *** *** SYTE (\$PAGE *** NOT** - 10, *** *** LENGTH (HEADER(\$NAHE 'INDEX\$))*	#FOR I = 0, 1, HEAD**NO \$ *********************************	#FOR I = 0, 1, HEAD*NO \$ "  ***BYTE(\$PAGE****IDTH - 10, 5\$) (HEADE*(\$\$E\$)) = 5H(PAGE*)  **********************************	FOR I = 0, 1, HEAD'NO S ""  ***SYTE(SPAGE'WIDTH - 10, 1, MARE'BYTE = MAKGIN + 1 & LENGTH(HEADER(SNAME'INDEXS))  ***AMARE'BYTE = MAKGIN + 1 & LENGTH(HEADER(SNAME'INDEXS))  ***AMARE'BYTE = MAKGIN + 1 & LENGTH(HEADER(SNAME'INDEXS))  ***AMARE'BYTE = MAKGIN + 1 & LENGTH(LENGTH)  **AMARE'BYTE = MA	FOR I = 0, 1, HEAD**NO \$ **	#FOR I = 0, 1, HEAD"NO \$ "  #################################
FOR I = 0, 1, HEAD'NO \$ "  FOR I = 0, 1, HEAD'NO \$ "  BYTE (\$PAGE'NIDTH - 10,  S\$) (HEADER(\$G\$)) = 5H(PAGE)  NAKE BYTE = HARGIN +  LENGTH(HEADER(\$NAME'INDEX\$))  + 1 \$	FOR I = 0.1, HEAD'NO \$ "  BYTE (SPAGE'NIOTH - 10.  \$ \$ \$ (HEADER(\$&\$)) = 5H(PAGE)  "NAKE'BYTE = MAKGIN +  "LENGTH (HEADER(\$MAME'INDEX\$))  " 1 \$  " 2 \$ "  " 2 * " " " " " " " " " " " " " " " " "	*FOR I = 0, 1, HEAD**NO \$ *****  *****************************	FOR I = 0, 1, HEAD**NO \$ *********************************	FOR I = 0, 1, HEAD'NO \$ ****  ******************************	FOR I = 0, 1, HEAD"NO \$ "  "BYTE (SPACE "MIDTH - 10,  "S (HEADER (SGS)) = 5H(PAGE)  "NAKE "BYTE = MAKGIN +  " LENGTH(HEADER (SNAME "INDEXS))  " 1	FOR I = 0, 1, HEAD'NO \$ ****  ******************************	FOR I = 0, 1, HEAD"NO \$ "  **SYTE (SPAGE WIDTH - 10,  **SYTE (SPAGE NIDTH - 10,  **SYTE (SPAGE NIDH	*FOR I = 0, 1, HEAD**NO \$ **  ***SYTE(\$FAGE**NIDIM - 10,	FOR I = 0, 1, HEAD"NO \$ "  ***SYTE (\$ PAGE "M DIM - 10,  ***S\$ (HEADER(\$ L\$) = 5H(PAGE )  ****AME" BYTE = MARGIN +  ****LENGTH(HEADER(\$ NAME" INDEX\$))  **********************************	FOR I = 0, 1, HEAD"NO \$ "  BYTE (\$PAGE WIDTH - 10,  \$ \$ (HEADER(\$&\$)) = 5H(PAGE)  "NAKE BYTE = MARGIN +  "LENGTH (HEADER(\$NAME "INDEX\$))  " LENGTH (HEADER(\$NAME "INDEX\$))	FOR I = 0, 1, HEAD NO S PRINCE SENTING S PRINCE S	FOR I = 0, 1, HEAD"NO \$ "  ***SYTE (\$PAGE "HIDTH - 10,  ***SYTE "HIDTH - 10,  **SYTE "HIDTH - 10,  ***SYTE "HIDTH - 10	*FOR I = 0, 1, HEAD**NO \$ **  ***SYTE(\$PAGE**NIDIM - 10, \$ ***  ***SYTE(\$FAGE**NIDIM - 10, \$ ***  ***SYTE(\$FAGE**NIDIM - 10, \$ ***  ****SYTE(\$FAGE**NIDIM - 10, \$ ***  ****SYTE	FOR I = 0, 1, HEAD"NO \$ "  BYTE (EPAGE MIDTH - 10,  S\$) (HEADER(S&3) = 5H(PAGE)  NAKE BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))
FOR I = 0, 1, HEAD'NO S ****  ******************************	FOR I = 0, 1, HCAD NO S  BYTE (SPAGE MIDTA - 10,  SS) (HEADER(\$6\$)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(\$NAME 'INDEXS))  + 1	FOR I = 0, 1, HcAD'NO \$ "  * FOR I = 0, 1, HcAD'NO \$ "  * BYTE (\$PAGE 'M IDIH - 10,  * \$ \$ (HEADER(\$E)) = 5H(PAGE )  * * * * * * * * * * * * * * * * * *	FOR I = 0, 1, HEAD'NO S ""  ********************************	FOR I = 0, 1, HEAD'NO \$ *  *** *** *** *** *** *** *** *** *	FOR I = 0, 1, HEAD"NO \$ "  *** SYTE (SPACE WIDTH - 10,  ** SANTE (SAS) = 5H(PAGE )  *** NAKE BYTE = MARGIN +  *** LENGTH HEADER(SNAHE 'INDEXS))  *** *** *** *** *** *** *** *** ***	FOR I = 0, 1, HEAD'NO \$ *  *** *** *** *** *** *** *** *** *	FOR I = 0, 1, HEAD'NO \$ ****  ******************************	FOR I = 0, 1, HEAD'NO \$ "  * FOR I = 0, 1, HEAD'NO \$ "  * BYTE (\$PAGE WIDTH - 10,  * \$ \$ (HEADER(\$E)) = 5H(PAGE )  * LENGTH (HEADER(\$NAME 'INDEX\$))  * LENGTH (HEADER(\$NAME 'INDEX\$))  * LENGTH (HEADER(\$NAME 'INDEX\$))  * I \$ "	FOR I = 0, 1, Héadono s  BYTE (EPAGE MIDTH - 10,  SS) (HEADER(SES) = 5H(PAGE)  NAME BYTE = MARGIN +  LENGTH (HEADER(SNAME INDEXS))  LENGTH (HEADER(SNAME INDEXS))  LENGTH (HEADER(SNAME INDEXS))  LENGTH (HEADER(SNAME INDEXS))  ITLE SN SFOR I = 0, 1, TITLE NO S  ITLE SN S	FOR I = 0, 1, HCAD NO S  BYTE (SPAGE MIDTH - 10,  SS (HEADER(SES)) = 5HPAGE )  NAME BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))  LENGTH (HEADER(SNA	FOR I = 0, 1, HCAD****  *******************************	FOR I = 0, 1, HEAD'NO \$ *  *** SYTE (\$ CACE 'N LOTH - 10,   ***	FOR I = 0, 1, Héadono &  * FOR I = 0, 1, Héadono &  * S\$ (Heade width - 10,  * S\$ (Heade width - 10,  * S\$ (Heade width - 10,  * Length (Heade Kishame indexs))  * Length (Heade Kishame indexs))  * Length (Heade Kishame indexs))  * ILLE SW &	FOR I = 0, 1, HCAD NO S  BYTE (SPAGE MIDTA - 10,  SS) (HEADER(\$6\$)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(\$NAME INDEXS))  TILE SM SFOR I = 0, 1, TITLE NO S  ITLE SM SFOR I = 0, 1, TITLE NO S  I = 0, 1, OUT BUF SIZE - 1 * ********************************
**************************************	FOR I = 0, 1, HEAD'NO \$ "  BYTE (\$PAGE'WIDTM - 10,  \$ 5\$) (HEADER(\$6\$)) = 5H(PAGE)  **NAKE'BYTE = MAKGIN +  **LENGTH(HEADER(\$NAME'INDEX\$))  **********************************	FOR I = 0, 1, HEAD NO S  BYTE (SPAGE WIDTH - 10,  SS) (HEADER(SEN) = 5H(PAGE)  NAME BYTE = MARGIN +  LENGTH (HEADER(SNAME INDEXS))	FOR I = 0, 1, HEAD'NO S ""  ********************************	FOR I = 0, 1, HCAD****  *******************************	FOR I = 0, 1, HEAD'NO S ""  ********************************	FOR I = 0, 1, HCAD****  *******************************	FOR I = 0, 1, HCAD****  *******************************	FOR I = 0, 1, HEAD NO S  BYTE (SPAGE WIDTH - 10,  SS) (HEADER(SES) = 5H(PAGE)  NAKE BYTE = MARGIN +  LENGTH (HEADER(SNAME INDEXS))  LENGTH (HEADER(SNAME INDEXS))  LENGTH (HEADER(SNAME INDEXS))  LENGTH (HEADER(SNAME INDEXS))  ITLESS S	FOR I = 0, 1, HEAD NO S  BYTE (SPAGE MIDTH - 10,  SS (HEADER(SES)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))  LLESS SFOR I = 0, 1, TITLE NO S  ITLE SN SFOR	FOR I = 0, 1, HEAD'NO S ""  ********************************	FOR I = 0, 1, HEAD'NO \$ "  *********************************	FOR I = 0, 1, HCAD'NO \$  *** FOR I = 0, 1, HCAD'NO \$  *** SYTE (\$ LADER (\$ LADER) = 5H(PAGE )  *** SANAKE BYTE = MAKGIN +  *** LENGTH (HEADER (\$ NAHE' INDEX\$))  *** LENGTH (HEADER (\$ NAHE' INDEX\$))  *** LENGTH (HEADER (\$ NAHE' INDEX\$))  *** LADER (\$ NAHE' INDEX\$)	FOR I = 0, 1, HEAD NO S  BYTE (EPAGE MIDTH - 10,  SS) (HEADER(SAS)) = 5H(PAGE)  NAME BYTE = MARGIN +  LENGTH (HEADER(SNAME INDEXS))	FOR I = 0, 1, HEAD NO S  BYTE (SPAGE WIDTH - 10,  SS) (HEADER(\$6\$)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(\$NAME INDEXS))  TILE SM SFOR I = 0, 1, TITLE NO S  ITLE SM SFOR I = 0, 1, TITLE NO S  I = 0, 1, OUT BUF SIZE - 1 * ********************************
# FOR I = 0, 1, HEAD*NO \$ *****  # BYTE (\$PAGE** MIDIA - 10,	FOR I = 0, 1, HEAD'NO S  BYTE (SPAGE'WIDTH - 10,  53) (HEADER(\$6\$)) = 5H(PAGE)  NAKE BYTE = MAKGIN +  LENGTH(HEADER(SNAME'INDEXS))  + 1 S  FOR I = 0, 1, TILE'NO S  FOR I = 0, 1, TILE'NO S	FOR I = 0, 1, HEAD'NO \$  BYTE (SPAGE MIDTA - 10,  \$ 58) (HEADER(\$68)) = 5H(PAGE)  **NAHE BYTE = MAKGIN +  **LENGTH (HEADER(\$NAHE * INDEX\$))  **1 \$  **ILE SN \$ **********************************	FOR I = 0, 1, HEAD'NO S P  *** SYTE (SPAGE MIDTA - 10, SS) (HEADER(SS)) = 5H(PAGE)  *** SYTE SYTE = MAKGIN + 10, SS) (HEADER(SNAME' INDEXS))  *** LENGTH (HEADER(SNAME' INDEXS)  ** LENGTH (HEADER(SNAME' INDEXS)  *** LENGTH (HEADER(SNAME' INDEXS	FOR I = 0, 1, HEAD'NO \$ ""  ***SYTE(\$PAGE ***IDTM = 10, 5, \$\$) (HEADER(\$6\$)) = 5H(PAGE)  ***NAKE BYTE = MAKGIN + LENGTH(HEADER(\$NAME ***INDEX\$))  ***LENGTH(HEADER(\$NAME ***INDEX\$))	FOR I = 0, 1, HEAD'NO S P  ***SYTE (SPAGE WIDTH - 10, SS) (HEADER(SS)) = 5H(PAGE)  ***AKE BYTE = MAKGIN + 1	FOR I = 0, 1, HEAD'NO S "	FOR I = 0, 1, HEAD'NO \$ ""  ***SYTE(\$PAGE ***IDTM - 10, 5 ***)  ***NAKE BYTE = MAKGIN + LENGTH(HEADER(\$NAME ***))  ***LENGTH(HEADER(\$NAME ***))	#FOR I = 0, 1, HEAD'NO \$ ""  **BYTE (\$PAGE "MIDTH - 10, 55) (HEADER(\$6\$)) = 5H(PAGE)  **NAHE BYTE = MAKGIN + LENGTH (HEADER(\$NAHE "INDEX\$))  **LENGTH (HEADER(\$NAHE "INDEX\$))	FOR I = 0, 1, HEAD'NO S PROPERTY OF STATE	FOR I = 0, 1, HEAD'NO S  ***********************************	FOR I = 0, 1, HEAD'NO S ****  ******************************	FOR I = 0, 1, HEAD'NO S ****  ******************************	#FOR I = 0, 1, HEAD'NO \$ ""  ***BYTE (\$PAGE "MIDTH - 10, 55) (HEADER(\$6\$)) = 5H(PAGE)  ***NAME BYTE = MAKGIN + LENGTH (HEADER(\$NAME "INDEX\$))  ***LENGTH (HEADER(\$NAME "INDEX\$))	FOR I = 0, 1, HEAD'NO S ****  ******************************
# FOR I = 0, 1, HEAD'NO \$ "  # FOR I = 0, 1, HEAD'NO \$ "  # BYTE (\$PAGE'WIDTH - 10, 8	FOR I = 0.1, HEAD'NO S PROPERTY OF STATE SPACE WIDTH - 10.5 S) (HEADER(\$6\$)) = 5H(PAGE) PROPERTY OF STATE SAME SAME SAME SAME SAME SAME SAME SAM	#FOR I = 0, 1, HEAD'NO \$ ""  ################################	#FOR I = 0, 1, HEAD'NO \$ "	FOR I = 0, 1, HEAD'NO S ***  ***  ***  ***  ***  ***  ***  *	FOR I = 0, 1, HEAD'NO S P  ***SYTE (SPAGE WIDTH - 10, SS) (HEADER(SS)) = 5H(PAGE)  ***AKE BYTE = MAKGIN + 1	FOR I = 0, 1, HEAD'NO S ****  ******************************	FOR I = 0, 1, HEAD'NO S PROPERTY OF STATE	### ### ##############################	FOR I = 0, 1, HEAD'NO S PROPERTY OF STATE SPACE WIDTH - 10, 55) (HEADER(\$6\$)) = 5H(PAGE) STATE SPACE S	FOR I = 0, 1, HEAD'NO S ****  ******************************	FOR I = 0, 1, HEAD'NO S PROPERTY OF THE CADER (SGS) = 5H(PAGE) = 5	FOR I = 0, 1, HEAD'NO S ****  ******************************	FOR I = 0, 1, HEAD'NO S ""  ***BYTE(\$PAGE'WIOTH - 10, 55) (HEADER(\$6\$)) = 5H(PAGE)  ***NAKE'BYTE = MAKGIN + LENGTH(HEADER(\$NAME'INDEXS))  ***LENGTH(HEADER(\$NAME'INDEXS))  ***LENGTH(HEADER(\$NAME'INDEXS)  ***LENGTH(HEADER(\$NAME'INDEXS))  ***LENGTH(HEADER(\$NAME'INDEXS)  ***LENGTH(HEADER(\$NAME'INDEXS)  ***LENGTH(HEADER(\$NAME'INDE	### ##################################
FOR I = 0, 1, HEAD'NO \$  BYTE(\$PAGE'NIDIM - 10,  \$ \$ (HEADER(\$G\$)) = 5H(PAGE)  LENGTH(HEADER(\$NAME'INDEX\$))  1 \$	FOR I = 0, 1, HEAD'NO \$ ***  ***BYTE (\$PAGE'MIDTM - 10, 55) (HEADER(\$G\$)) = 5H(PAGE)  ***********************************	FOR I = 0, 1, HCAD'NO S  BYTE (SPAGE MIDTH - 10, 5 S) (HCADER (SUS)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HCADER (SNAME INDEXS))  + 1 S  LILE'SN SFOR I = 0, 1, TITLE'NO S  I = 0, 1, OUT'BUF'SIZE - 1	FOR I = 0, 1, HEAD'NO \$ "  ***SYTE(\$PAGE WIDTM - 10, 8 * 1, 8 * 1)	FOR I = 0, 1, HEAD'NO \$ "  **SYTE(SPACE'NIDTH - 10, 1, 10, 10, 10, 10, 10, 10, 10, 10,	FOR I = 0, 1, HEAD'NO \$ "  ***SYTE(\$PAGE WIDTH - 10, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	FOR I = 0, 1, HEAD'NO \$ "  ***SYTE(\$FAGE ***IDTH - 10,	FOR I = 0, 1, HEAD'NO \$ "  ***SYTE(SPACE'NIDTH - 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,	FOR I = 0, 1, HEAD'NO S "  "BYTE (SPAGE'WIDTH - 10, 55) (HEADER(\$68)) = 5H(PAGE)  "NAKE BYTE = MAKGIN +  "LENGTH (HEADER(\$NAME'INDEXS))  "1	FOR I = 0, 1, HEAD'NO \$ "  ***AFE ***PTE *** HANGE **  ***AFE ***PTE *** HANGE **  ***AFE ***PTE ***  ***AFE **  ***AFE ***  **AFE ***  ***AFE ***  **AFE **  **AFE ***  **AFE **  **AFE	FOR I = 0, 1, HEAD'NO \$ ***  *******************************	FOR I = 0, 1, HEAD'NO \$ "  ***SYTE(\$FAGE WIDTH - 10, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	FOR I = 0, 1, HEAD'NO \$ "  ***SYTE(\$FAGE ***IDTH - 10, 1, 10, 10, 10, 10, 10, 10, 10, 10,	FOR I = 0, 1, HEAD'NO \$ "  "BYTE (\$PAGE'WIDTH - 10,  "S\$) (HEADER(\$6\$)) = 5H(PAGE)  "NAKE BYTE = MAKGIN +  "LENGTH(HEADER(\$NAME'INDEX\$))  "1	FOR I = 0, 1, HEAD'NO \$ "  *********************************
FOR I = 0, 1, HEAD'NO \$  *******************************	FOR I = 0. 1. HEAD'NO S ****  ******************************	FOR I = 0, 1, HCAD*NO \$ " FOR I = 0, 1, HCAD*NO \$ " FOR I = 0, 1, HCAD*NO \$ " FOR I = 0, 1, TITLE*NO \$ " FOR I	FOR I = 0.1, HcAD*NO \$  ***SYTE(SPAGE***MIDTH** - 10.*  ***SYTE(SPAGE****MIDTH*** - 10.*  ***SYTE(SPAGE************************************	FOR I = 0, 1, HCAD*NO \$  ***SYTE(SPAGE****IDTM - 10, 1, MCAD************************************	FOR I = 0.1, HcAD*NO \$  ***SYTE(SPAGE***MIDTH** - 10.*  ***SYTE(SPAGE***MIDTH*** - 10.*  ***SYTE(SPAGE***MIDTH***MIDTH***MIDTH***MIDTH**  ***SYTE(SPAGE***MIDTH***MIDTH***MIDTH***MIDTH**  ***SYTE(SPAGE***MIDTH**MIDTH***MIDTH**MIDTH**  ***SYTE(SPAGE***MIDTH**MIDTH**MIDTH**MIDTH**  ***SYTE(SPAGE***MIDTH**MI	FOR I = 0, 1, HCAD*NO \$  ***SYTE(SPAGE****IDTM - 10, 1, MCAD************************************	FOR I = 0, 1, HCAD NO S  BYTE (SPAGE MIDTH - 10,  SS (HCADER (SGS)) = 5H(PAGE)  S NAKE BYTE = MAKGIN +  LENGTH (HEADER (SNAME INDEXS))  + 1 S  LILE'SN SFOR I = 0, 1, TITLE'NO S  LITLE'SN SFOR I = 0, 1, TITLE'NO S	FOR I = 0.1, HCAD NO S  *** *** *** *** *** *** *** *** **	FOR I = 0. 1, HEAD'NO \$ *  *** FOR I = 0. 1, HEAD'NO \$ *  *** S\$) (HEADER(\$G\$)) = 5H(PAGE)  ** S\$) (HEADER(\$G\$)) = 5H(PAGE)  *** S\$) (HE	FOR I = 0. 1. H&AD'NO \$  ***SYTE(\$FAGE ***IDTM - 10.**  ***SYTE(\$FAGE ****IDTM - 10.**  **SYTE(\$FAGE ****IDTM - 10.**  ***SYTE(\$FAGE ****IDTM - 10.**  ***	FOR I = 0.1, HCADONO \$  ***SYTE(SPAGE MIDTH - 10.**  ***SYTE(SPAGE MI	FOR I = 0.1, HCAD***  ********************************	FOR I = 0.1, HEAD'NO \$ " FOR I = 0.1, HEAD'NO \$ " FOR I = 0.1, HEAD'NO \$ " FOR I = 0.1, TITLE NO \$ " FOR I = 0.1, TITLE	FOR I = 0. 1, H&AO'NO \$  *** *** *** *** *** *** *** *** **
FOR I = 0, 1, HEAD'NO \$ ""  ********************************	FOR I = 0, 1, HEAD'NO \$ ""  ***SYTE (\$ PAGE "MIDIM - 10, "  ***S\$ (HEADER(\$ L\$) = 5H(PAGE )  ***S********************************	FOR I = 0, 1, HEAD'NO S  ***STEERER FREE FREE FREE FREE FREE FREE FRE	FOR I = 0, 1, HEAD'NO S " "BYTE (SPAGE MIDTH - 10, "S\$) (HEADER (\$6.8)) = 5H(PAGE) "NAHE BYTE = MAKGIN + "LENGTH (HEADER (\$NAHE INDEX\$))" "1	FOR I = 0, 1, HEAD'NO S "  *** *** *** *** *** *** *** *** ***	FOR I = 0, 1, HEAD'NO S " "BYTE (SPAGE MIDTH - 10, "SS) (HEADER (SAS)) = 5H(PAGE) "NAHE BYTE = MAKGIN + "LENGTH (HEADER (SNAHE INDEXS))" "1	FOR I = 0, 1, HEAD'NO S "  *** *** *** *** *** *** *** *** ***	FOR I = 0, 1, HEAD'NO S "  *********************************	FOR I = 0, 1, HEAD'NO S "  ***SYTE(SPAGE'MIDTH - 10, S S) (HEADER(S&S)) = 5H(PAGE)  ***AAKE'BYTE = MAKGIN + 10, S S)  ***AAKE'BYTE = MAKGIN + 10, S S S)  ***AAKE'BYTE = MAKGIN + 10, S S S S S S S S S S S S S S S S S S S	FOR I = 0, 1, HEAD*NO S *****  *****************************	FOR I = 0.1, HEAD'NO \$ " FOR I = 0.1, HEAD'NO \$ " BYTE(SPAGE'MIDTH - 10.  S S (HEADER(SAS)) = 5H(PAGE)  S NAKE BYTE = MAKGIN +  LENGTH (HEADER(SNAME'INDEXS))  1 LENGTH (HEADER(SNAME'INDEXS))	FOR I = 0, 1, HEAD'NO \$ "  *********************************	FOR I = 0, 1, HEAD'NO S "  *** SYTE (\$PAGE "MIDTH - 10, 1, MARE" BYTE (\$PAGE "MIDTH - 10, 1, MARE" BYTE = MAKGIN + 1	FOR I = 0, 1, HEAD'NO S ""  ***SYTE(SPAGE "MIDTH - 10, 1 **  ***SYTE(SPAGE "MAKE") = 5H(PAGE)	FOR I = 0.1, HEAD'NO \$ " FOR I = 0.1, HEAD'NO \$ " BYTE (EPAGE MIDTH - 10.  S
FOR I = 0, 1, HEAD'NO S *  BYTE (SPAGE ** NOTH - 10,  SS) (HEADER(SGS)) = 5H(PAGE)*  ** NAKE BYTE = MARGIN +  ** LENGTH(HEADER(SNAME*INDEX*))*  ** 1 S  ** ** 1 S	FOR I = 0, 1, HEAD'NO \$ ***  * FOR I = 0, 1, HEAD'NO \$ ***  * SYTE (SPAGE *** MIDIH - 10, ***  * SYTE (SPAGE ***  * NAME BYTE = MAKGIN +	FOR I = 0, 1, HEAD"NO \$  **SYTE (\$CAC ** ** ** ** ** ** ** ** ** ** ** ** **	FOR I = 0, 1, HEAD*NO \$ **********************************	FOR I = 0, 1, HEAD"NO \$ ""  ***SYTE (\$PAGE "HIDTH - 10, " SS) (HEADER(\$6.8)) = 5H(PAGE)  ***AKE BYTE = MAKGIN + 10, " SS) (HEADER(\$NAME "INDEXS)) + 1	FOR I = 0, 1, HEAD*NO \$ **********************************	FOR I = 0, 1, HEAD"NO \$ ""  ***SYTE (\$PAGE "MIDTH - 10, " SS) (HEADER(\$6.8)) = 5H(PAGE)  ***AKE BYTE = MAKGIN + 10, " SS) (HEADER(\$NAME "INDEXS)) + 1	FOR I = 0, 1, HEAD"NO \$ ""  ***SYTE (SPAGE WIDTH - 10,	FOR I = 0, 1, HEAD"NO \$  **SYTE(SPAGE WIDTH - 10,  **SS)(HEADER(SGS)) = 5H(PAGE )  **NAKE BYTE = MAKGIN +  **LENGTH(HEADER(SNAHE INDEXS))	#FOR I = 0, 1, HEAD**NO \$ ****  ******************************	FOR I = 0, 1, HEAD"NO \$ "  BYTE (\$PAGE WIDTH - 10,  \$ \$ (HEADER(\$&\$)) = 5H(PAGE)  **NAKE BYTE = MARGIN +  **LENGTH (HEADER(\$MAME "INDEX\$))  **ILE*SM \$ "*FOR I = 0, 1, TITLE*NO \$ "  ITLE*SM \$ "*F	FOR I = 0, 1, HEAD'NO \$	FOR I = 0, 1, HEAD"NO \$ ""  ***SYTE(SPAGE WIDTH - 10, 1, MARE BYTE SANGIN + 10, 1, MARE BYTE = MARGIN + 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	FOR I = 0, 1, HEAD"NO \$ "  ***SYTE (\$ C = 0,	FOR I = 0, 1, HEAD"NO \$ "  * FOR I = 0, 1, HEAD"NO \$ "  * S\$ (HEADER(\$&\$1) = 5H(PAGE)  * NAKE BYTE = MAKGIN +  * LENGTH(HEADER(\$NAME.INDEX\$))  * 1 \$  * ITLE*SN \$ "FOR I = 0, 1, TITLE*NO \$ "  * ITLE*SN \$ "FOR I = 0, 1, TITLE*NO \$ "  * ITLE*SN \$ "FOR I = 0, 1, TITLE*NO \$ "  * ITLE*SN \$ "FOR I = 0, 1, TITLE*NO \$ "  * ITLE*SN \$ "FOR I = 0, 1, TITLE*NO \$ "  * ITLE*SN \$ "FOR I = 0, 1, TITLE*NO \$ "  * ITLE*SN \$ "FOR I = 0, 1, TITLE*NO \$ "  * ITLE*SN \$ "FOR I = 0, 1, TITLE*NO \$ "  * ITLE*SN \$ "FOR I = 0, 1, TITLE*NO \$ "  * ITLE*SN \$ "FOR I = 0, 1, TITLE*NO \$ "  * ITLE*SN \$ "FOR I = 0, 1, TITLE*NO \$ "  * ITLE**NO \$ "FOR I = 0, 1, TITLE*NO \$ "  * ITLE**NO \$ "FOR I = 0, 1, TITLE**NO \$ "  *
FOR I = 0, 1, HEAD'NO S FOR EAST OF STATE OF STA	FOR I = 0, 1, HCADONO & FOR I = 0, 1, HCADONO & BYTE (SPAGE MIDTA - 10,  52) (HEADER(\$62)) = 5H(PAGE)  **NAME BYTE = MAKGIN +  **LENGTH (HEADER(\$NAME **INDEX\$))  ** 1 & **INDEX\$)  ** 1 & **INDEX\$  ** 1 & **	FOR I = 0, 1, HEAD" NO \$ " FOR I = 0, 1, HEAD" NO \$ " BYTE (\$PAGE WIDTH - 10, \$ \$ (HEADER(\$E)) = 5H(PAGE) \$ \$ LENGTH (HEADER(\$NAME "INDEX\$)) \$ \$ \$ (ITLE SW \$ "FOR I = 0, 1, TITLE "NO \$ " ETTERS SW \$ "	FOR I = 0, 1, HEAD'NO \$  *** FOR I = 0, 1, HEAD'NO \$  *** SYTE (SPACE "MIDTH - 10, 1, MAKE" SYTE = MAKGIN + 10, 1, MAKE" SYTE = MAKGIN + 10, 1, MAKE" SYTE = MAKGIN + 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	FOR I = 0, 1, HEAD'NO \$ " FOR I = 0, 1, HEAD'NO \$ "  ***SYLE(\$FACE'NIDIT - 10,  ***SYLE(\$FACE'NIDIT - 10,  ***SYLE(\$FACE'NIDIT - 10,  ***SYLE(\$FACE'NIDIT - 10,  ***SYLE'SN \$ "FOR I = 0, 1, TITLE'NO \$ "  ***SYLE'SN \$ "	FOR I = 0, 1, HEAD"NO \$ "  ***SYTE (\$PAGE "HIDTH - 10, " SA) (#A) (#A) (#A) (#A) (#A) (#A) (#A) (#	FOR I = 0, 1, HEAD'NO \$ " FOR I = 0, 1, HEAD'NO \$ " Syldeadewill = 5H(PAGE)  **AAKE BYTE = MAKGIN + **LENGTH(HEADER(SNAHE'INDEXS)) **1	FOR I = 0, 1, HEAD'NO \$ " FOR I = 0, 1, HEAD'NO \$ "  BYTE (SAGE WIDTH - 10,  SS) (HEADER(SUS)) = 5H(PAGE)  LENGTH (HEADER(SWAHE'INDEXS))  LA 1	FOR I = 0, 1, HEAD" NO \$ " FOR I = 0, 1, HEAD" NO \$ " BWTE (\$AAGE WIDTH - 10, \$ \$ (HEADER(\$LS)) = 5H(PAGE) \$ **NAKE BYTE = MAKGIN + **LENGTH(HEADER(\$NAME INDEXS)) **LENGTH(HEADER(\$NAME INDEXS)) **LENGTH(HEADER(\$NAME INDEXS)) **LENGTH(HEADER(\$NAME INDEXS)) **LENGTH(HEADER(\$NAME INDEXS)) **LILE SW \$ "FOR I = 0, 1, TITLE NO \$ "	FOR I = 0, 1, HEAD'NO \$ " FOR I = 0, 1, HEAD'NO \$ " S\$) (HEADER(SES) = 5H(PAGE) \$ "NAKE BYTE = MARGIN +	FOR I = 0, 1, HEAD'NO \$ " FOR I = 0, 1, HEAD'NO \$ " BYTE (EPAGE MIDTH - 10,  S\$ (HEADER(S&B)) = 5H(PAGE)  "NAKE BYTE = MAKGIN +  "LENGTH (HEADER(SMAME INDEXS))  " 1	FOR I = 0, 1, HCAD NO S FOR I = 0, 1, HCAD NO S BYTE (SPAGE MIDTH - 10, 55) (HCADER(\$6\$)) = 5H(PAGE)  **NAKE BYTE = MAKGIN + **LENGTH (HCADER(\$NAME INDEXS)) ***********************************	FOR I = 0, 1, HEAD'NO \$ ***  ***SYTE (SPAGE WIDTH - 10, **	FOR I = 0, 1, HEAD'NO \$ " FOR I = 0, 1, HEAD'NO \$ " BYTE (\$PAGE WIDTH - 10, \$ \$ (HEADER(\$E)) = 5H(PAGE) \$ "NAKE BYTE = MARGIN + " LENGTH (HEADER(\$NAME "INDEX\$)) * 1 \$ " ITLE SW \$ "FOR I = 0, 1, ITLE "NO \$ "	FOR I = 0, 1, HCAD'NO \$ " FOR I = 0, 1, HCAD'NO \$ " BYTE (\$PAGE MIDTM = 10, \$ \$ (HEADER(\$6.8)) = 5H(PAGE)  **NAHE BYTE = MAKGIN + **LENGTH (HEADER(\$NAHE "INDEX\$)) ***********************************
PAGE BYTE = PAGE MIDIN - 5 8	PAGE BYTE = PAGE NIUIN - 58 **  ********************************	FOR I = 0, 1, HCAD NO S	PAGE BYTE = PAGE NIUTH - 5 8 - 1	PAGE BYTE = PAGE WIDIN - 58 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	PAGE BYTE = PAGE WIDTH - 58 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	PAGE BYTE = PAGE WIDIN - 58 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	PAGE BYTE = PAGE WIDIN - 58 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	FOR I = 0, 1, HCAD NO S  BYTE (SPAGE WIDTH - 10,  BYTE (SPAGE WIDTH - 10,  S (HEADER(SES)) = 5H(PAGE)  NAME BYTE = MARGIN +  LENGTH (HEADER(SNAME INDEXS))  LENGTH (HEADER(SNAME INDEXS))  LENGTH (HEADER(SNAME INDEXS))  IT = 0, 1, OUT BUF SIZE - 1 **********************************	FOR I = 0, 1, HCAD'NO \$ "	PAGE BYTE = PAGE WIDIN - 58 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	PAGE BYTE = PAGE WIDTH - 58 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	PAGE BYTE = PAGE WIDIN - 5 8 FOR I = 0, 1, HEAD'NO 8 BYTE (\$AAGE WIDIN - 10, 8 S) (HEADER(\$LS)) = 5H(PAGE) (LENGTH (HEADER(\$LS)) = 1, TITLE'NO 8 S' FOR I = 0, 1, TI	FOR I = 0, 1, HCAD NO S  SYTE (SPAGE MIDTH - 10,  SS) (HEADER(SES)) = 5H(PAGE)  NAME BYTE = MARGIN +  LENGTH (HEADER(SNAME INDEXS))	PAGE BYTE = PAGE WIDIN - 5 S  FOR I = 0, 1, HEAD NO S  BYTE (SPAGE WIDIM - 10,  S 5 S) (HEADER(\$6.8)) = 5H(PAGE)  S NAKE BYTE = MAKGIN +  LENGTH(HEADER(\$NAME INDEXS))  F 1 S  F
PAGE BYTE = PAGE MIDTH - 5 8  FOR I = 0, 1, HEAD NO 8 PROPERTY OF STATE OF	**************************************	FOR I = 0,1, HEAD NO S PROFESSION S PROFESSI	PAGE BYTE = PAGE NIDTH - 5 8  FOR I = 0, 1, HEAD'NO 8 P  BYTE (SPAGE WIDTH - 10,  58) (HEADER(SS)) = 5H(PAGE)  * AAKE BYTE = MAKCIN +  * LENGTH (HEADER(SNAME 'INDEXS))  * 1	PAGE BYTE = PAGE NIDTH - 5 8  FOR I = 0, 1, HEAD BY S  BYTE (\$PAGE MIDTH - 10, S  S  NAME BYTE = MARGIN +  LENGTH (HEADER(SMAME INDEXS))  1 = 0, 1, OUT BUF SIZE - 1 OUT LINE(SET)	PAGE BYTE = PAGE NIDTH - 5 8  FOR I = 0, 1, HEAD'NO 8 P  BYTE (\$PAGE WIDTH - 10,  5 \$) (HEADER(\$S)) = 5H(PAGE)  ***AKE********************************	PAGE BYTE = PAGE NIDTH - 5 8  FOR I = 0, 1, HEAD BY	PAGE BYTE = PAGE WIDTH - 5 8  FOR I = 0, 1, HEAD BY S  BYTE (\$PAGE WIDTH - 10, S  S  NAME BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))	FOR I = 0, 1, HEAD NO & " FOR I = 0, 1, HEAD NO & " FOR I = 0, 1, HEAD NO & " FOR I = 0, 1, HEAD NO & " FOR I = 0, 1, HEAD NO & " FOR I = 0, 1, IILE NO & " FOR I = 0, IILE NO	FOR I = 0, 1, HEAD NO & V FOR I = 0, 1, HEAD NO & V FOR I = 0, 1, HEAD NO & V FOR I = 0, 1, HEAD NO & V FOR I = 0, 1, HEAD NO & V FOR I = 0, 1, IIILE NO & V FOR I = 0, IILE NO & V FOR I =	PAGE BYTE = PAGE WIDTH - 5 8  FOR I = 0, 1, HEAD NO 8  BYTE (\$PAGE WIDTH - 10,  S\$) (HEADER(\$6\$)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(\$NAME INDEXS))  TILE SM \$ **********************************	PAGE BYTE = PAGE WIDTH - 5 8  FOR I = 0, 1, HEAD'NO 8 P  BYTE (SPAGE WIDTH - 10,  52) (HEADER(\$63)) = 5H(PAGE)  ***  ***  ***  ***  ***  ***  ***	PAGE BYTE = PAGE NIDTH - 5 8  FOR I = 0,1, HEAD NO 8  ***********************************	PAGE BYTE = PAGE WIDTH - 5 8  FOR I = 0, 1, HEAD NO 8  BYTE (\$PAGE WIDTH - 10,  \$ \$ (HEADER(\$\$)) = 5H(PAGE)  "NAME BYTE = MAKGIN +  "LENGTH (HEADER(\$NAME "NDEX\$))  " 1	PAGE BYTE = PAGE NIDTH - 5 8  FOR I = 0, 1, HEAD'NO \$ "  BYTE (\$PAGE WIDTH - 10,  5 \$) (HEADER(\$63)) = 5H(PAGE)  8 NAKE BYTE = MAKGIN +  LENGTH(HEADER(\$NAME INDEXS))  4 1 \$  TILE SM \$ "FOR I = 0, 1, TITLE NO \$ "  ITLE SM \$ "FOR I =
PAGE BYTE = PAGE NIDTH - 5 8 **  ** ** ** ** ** ** ** ** ** ** ** **	**************************************	FOR I = 0, 1, HEAD'NO \$  "FOR I = 0, 1, HEAD'NO \$  "FOR I = 0, 1, HEAD'NO \$  "SS) (HEADE (SES)) = 5H(PAGE)  "NAME BYTE = MAKGIN +  "LENGTH (HEADER(SNAME INDEXS))  "A S S S S S S S S S S S S S S S S S S S	PAGE STEE = PAGE NIDTH - 5 S FOR I = 0, 1, HEAD'NO S PROPERTY OF STEE STEE STEE STEE STEE STEE STEE STE	FOR I = 0, 1, HEAD'NO \$ "  *********************************	PAGE STEE = PAGE NIDTH - 5 S  FOR I = 0, 1, HEAD'NO S P  SYTE (SPAGE WIDTH - 10,  Syl (HEADER(SS)) = 5H(PAGE)  NAKE BYTE = MAKGIN +  LENGTH (HEADER(SNAHE'INDEXS))  1 I = 0, 1, OUT'BUF'SIZE - 1  THE SYLETTERS OF THE STATE OF	PAGE BYTE = PAGE NIDTH - 5 8  FOR I = 0, 1, HEAD NO 8  BYTE (\$PAGE NIDTH - 10, 8  S (HEADER(\$&)) = 5H(PAGE)  S (HEADER(\$&)) = 5H(	PAGE BYTE = PAGE NIDTH - 5 & FOR I = 0, 1, HEAD NO & FOR I = 0, 1, TILE NO & F	FOR I = 0, 1, HEAD'NO \$ "	PAGE STE = PAGE NIDIH - 5 S  FOR I = 0, 1, HEAD NO S  BYTE (SPAGE WIDTH - 10,  SS) (HEADER(SES)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))  + 1 S  LENGTH (HEADER(SNAME INDEXS))  + 1 S  LENGTH (HEADER(SNAME INDEXS))  ITLE SM SFOR I = 0, 1, TITLE NO S  ITLE SM SF	PAGE STE = PAGE NIDIH - 5 8  FOR I = 0, 1, HEAD'NO 8 P  BYTE (SPAGE WIDTH - 10,  SS) (HEADER(SAS)) = 5H(PAGE)  SNAKE BYTE = MAKGIN +  LENGTH(HEADER(SNAME INDEXS))  + 1 8  TILE'SM 8 PFOR I = 0, 1, TITLE'NO 8 P  ETTESM 8 PFOR I	PAGE STEE = PAGE NIDTH - 5 S  FOR I = 0, 1, HEAD'NO S P  STEE STEE STEE MACIN +  LENGTH HEADER(SCS)) = 5H(PAGE)  ***  ***  ***  ***  ***  ***  ***	FOR I = 0, 1, HEAD'NO S PROFESSED S S S S S S S S S S S S S S S S S	PAGE STTE = PAGE NIDTH - 5 S  FOR I = 0, 1, HEAD NO S  BYTE (SPAGE WIDTH - 10,  S S) (HEADER(\$6)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(\$NAME INDEXS))  TILE SM SFOR I = 0, 1, TITLE NO S F  ITLE SM SFOR I = 0, TITLE NO S F  ITLE SM SFOR I = 0, TITLE NO S F  ITLE SM SFOR I = 0, TITLE NO S F  ITLE SM SFOR I = 0, TITLE NO S F  ITLE SM SFOR I = 0, TITLE NO S F  ITLE SM SFOR I = 0, TITLE NO S F  ITLE SM SFOR I = 0, TITLE NO S F  ITLE SM SFOR I = 0, TITLE NO S F  ITLE SM SFOR I =	PAGE STEE = PAGE NIDTH - 5 S  FOR I = 0, 1, HEAD'NO S P  BYTE (SPAGE WIDTH - 10,  5 S) (HEADER(\$6.8)) = 5H(PAGE)  ***AKE BYTE = MAKGIN +  **LENGTH(HEADER(\$NAME INDEXS))  ***AKE BYTE = MAKGIN +  ***LENGTH(HEADER(\$NAME INDEXS))  ***AKE BYTE = MAKGIN +  ***A
PAGE BYTE = PAGE NIDTH - 5 8 **  *******************************	**************************************	PAGE BAGE PAGE NIDTH - 5 8  FOR I = 0, 1, HEAD NO 8  BYTE (SPAGE WIDTH - 10,  SS) (HEADER(\$6\$)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(\$NAME INDEX\$))  LENGTH (HEADER(\$NAME INDEX\$))  LILE SW 8FOR I = 0, 1, TITLE NO 8  LENGTH (HEADER(\$NAME INDEX\$))	**************************************	FOR I = 0, 1, HEAD'NO S * FOR I = 0, 1, TILLE'NO S * FOR I = 0, 1	FOR I = 0, 1, HEAD'NO S PACE'NE STATE STAT	FOR I = 0, 1, HEAD'NO S * FOR I = 0, 1, HILE'NO S * FOR I = 0, 1, TILLE'NO S *	PAGE BAGE PAGE NIDTH - 5 8  FOR I = 0, 1, HEAD'NO 8  BYTE (SPAGE MIDTH - 10, 1, HEAD'NO 8  SS) (HEADER(\$6\$)) = 5H(PAGE)  SNAHE BYTE = MAKGIN +  LENGTH (HEADER(\$NAHE 'INDEX\$))	PAGE BYTE = PAGE NIDTH - 5 8  FOR I = 0, 1, HEAD NO 8 P  BYTE (SPAGE WIDTH - 10,  SS) (HEADER(SES)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH (HEADER(SNAME INDEXS))  * 1	PAGE BYTE = PAGE NIDTH - 5 8 ***  ******************************	PAGE BYTE = PAGE NIDTH - 5 8 ***  ******************************	**************************************	FOR I = 0, 1, HEAD'NO S *  *******************************	PAGE BYTE = PAGE NIDTH - 5 8  FOR I = 0, 1, HEAD NO 8 PROPERTY STATE (\$PAGE WIDTH - 10, 1, HEAD NO 8 PROPERTY STATE (\$PAGE WIDTH - 10, 1, HEAD NO 8 PROPERTY STATE (\$PAGE NIDTH - 10, 1, HEAD NO 8 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 11 TLE NO 8 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 11 TLE NO 8 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 001 BUF SIZE - 1 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 001 BUF SIZE - 1 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 001 BUF SIZE - 1 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 001 BUF SIZE - 1 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 001 BUF SIZE - 1 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 001 BUF SIZE - 1 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 001 BUF SIZE - 1 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 001 BUF SIZE - 1 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 001 BUF SIZE - 1 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 001 BUF SIZE - 1 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 001 BUF SIZE - 1 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 001 BUF SIZE - 1 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 001 BUF SIZE - 1 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 001 BUF SIZE - 1 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 001 BUF SIZE - 1 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 001 BUF SIZE - 1 PROPERTY STATE (\$PAGE NIDTH - 10, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	PAGE BYTE = PAGE NIDTH - 5 8  FOR I = 0, 1, HEAD'NO 8 P  BYTE (\$PAGE WIDTH - 10,  53) (HEADER(\$S)) = 5H(PAGE)  *** A K *** BYTE *** BYTE *** BYTE ***  *** LENGTH(HEADER(\$NAME 'INDEX\$))  *** A K *** BYTE ***  *** LENGTH(HEADER(\$NAME 'INDEX\$))  *** A K ***  *** I * **  *** I * ***  *** I * ***  *** I * ***  *** I * ***  *** I * **  ** I * **  *** I * **  **
PAGE BYTE = PAGE *** MIDIM - 5 % ***  *** FOR I = 0, 1, HEAD** NO % **  *** FOR I = 0, 1, HEAD** NO % **  *** FOR I = 0, 1, HEAD** NO % **  *** FOR I = 0, 1, HEAD** NO % **  *** FOR I = 0, 1, HEAD** NO % **  *** FOR I = 0, 1, HEAD** NO % NO	*** FOR I = 0, 1, HCAD**NO \$ *****  *****************************	PAGE BYTE = PAGE MIDTH - S & FOR I = 0, 1, HEAD'NO & PERCENTRY AND S & PAGE WIDTH - 10, 5 & FOR I = 0, 1, HEAD'NO & PERCENTRY AND S & PERC	FOR I = 0, 1, HEAD'NO \$ " FOR I = 0, 1, TILE'NO \$ "	FOR I = 0.1, HEAD'NO & " FOR I = 0.1, TILE'NO & "	FOR I = 0, 1, HEAD'NO \$ " FOR I = 0, 1, TILE'NO \$ " FOR I = 0, 1, TI	FOR I = 0.1, HEAD'NO & " FOR I = 0.1, TILE'NO & "	FOR I = 0.1, HEAD'NO & " FOR I = 0.1, TILE'NO & " ITLE'SM & "FOR I = 0.1, TILE'SM & " ITLE'SM & "FOR I = 0.1, TILE'SM & " ITLE'SM & "FOR I = 0.1, TILE'SM & " ITLE'SM & "FOR I = 0.1, TILE'SM & " ITLE'SM & "FOR I = 0.1, TILE'SM & " ITLE'SM & "FOR I = 0.1, TILE'SM & " ITLE'SM & "FOR I = 0.1, TILE'SM & " ITLE'SM & "FOR I = 0.1, TILE'SM & " ITLE'SM & "FOR I = 0.1, TILE'SM & " ITLE'SM & "FOR I = 0.1, TILE'SM & "	PAGE BYTE = PAGE MIDTH - S & FOR I = 0, 1, HEAD'NO & PAGE WIDTH - 10, 5 S) (HEADER(SS)) = 5H(PAGE) & 5S) (HEADER(SS)) & 5	PAGE BYTE = PAGE MIDTH - 5 & FOR I = 0, 1, HEAD'NO & PAGE WIDTH - 10, 1, HEAD'NO & PAGE WIDTH -	FOR I = 0, 1, HEAD'NO S PACE BYTE (SPAGE WIDTH - 10, 1, HEAD'NO S PACE BYTE (SPAGE WIDTH - 10, 1, HEAD'NO S PACE BYTE (SPAGE WIDTH - 10, 1, HEAD'NO S PACE BYTE (SPAGE WIDTH - 10, 1, HEAD'NO S PACE BY SERVICE B	FOR I = 0, 1, HEAD'NO \$ *  ***FOR I = 0, 1, HEAD'NO \$ *  *******************************	FOR I = 0.1, HEAD'NO & " FOR I = 0.1, TILE'NO & "	PAGE BYTE = PAGE MIDTH - S & FOR I = 0, 1, HEAD NO & PAGE MIDTH - 10, 1, HEAD NO & PAGE MIDTH -	FOR I = 0, 1, HEAD'NO S **  ***SETTE HARGE N  ***SETTE HARGE N  ***SETTE HARGE N  ***SETTE HARGE N  ***SETTE HARGEN  ***SE
**************************************	**************************************	PAGE BYTE = PAGE WIDTH - 5 8 PERCENTERS PAGE WIDTH - 5 8 PERCENTERS PAGE WIDTH - 10 PERCENTERS PAGE PAGE PAGE PAGE PAGE PAGE PAGE PAGE	FOR I = 0, 1, HEAD'NO \$ "  FOR I = 0, 1, HEAD'NO \$ "  FOR I = 0, 1, HEAD'NO \$ "  FOR I = 0, 1, TILE NO \$ "  FOR I = 0, 1, TI	PAGE BYTE = PAGE WIDTH - S & PAGE WIDTH - S & PAGE WIDTH - S & PAGE WIDTH - 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10	FOR I = 0, 1, HEAD'NO \$  FOR I = 0, 1, HEAD'NO \$  FOR I = 0, 1, HEAD'NO \$  FOR I = 0, 1, TILE'NO \$  FOR I = 0, 1, OUT'BUF'SIZE - 1 **********************************	PAGE BYTE = PAGE WIDTH - S & PAGE WIDTH	PAGE BYTE = PAGE WIDTH - S & PAGE WIDTH - S & PAGE WIDTH - S & PAGE WIDTH - 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10	PAGE BYTE = PAGE WIDTH - 5 8 **  FOR I = 0, 1, D. HEAD NO 8 **  **  **  **  **  **  **  **  **  *	PAGE BYTE = PAGE WIDTH - 5 & PAGE WIDTH - 5 & PAGE WIDTH - 10	PAGE BYTE = PAGE WIDTH - 5 8 **  *******************************	PAGE BYTE = PAGE WIDTH - 5 8 **  *******************************	PAGE BYTE = PAGE WIDTH - S & PAGE WIDTH	PAGE BYTE = PAGE WIDTH - S & PAGE WIDTH - S & PAGE WIDTH - LO & PAGE WIDTH - P	PAGE BYTE = PAGE WIDTH - 5 8 **  *******************************
**************************************	*** *** *** *** *** *** *** *** *** **	**************************************	FOR I = 0, 1, HEAD'NO S "  *** FOR I = 0, 1, HEAD'NO S "  *** FOR I = 0, 1, HEAD'NO S "  *** FOR I = 0, 1, HEAD'NO S "  *** FOR I = 0, 1, TILE NO S "  *** I S "  *** I S "  *** I S "  *** I S "	PAGE DYTE = PAGE NIOTH - 5 & PAGE PAGE PAGE PAGE PAGE PAGE PAGE PAGE	FOR I = 0, 1, HEAD'NO S " FOR I = 0, 1, HEAD'NO S " FOR I = 0, 1, HEAD'NO S " FOR I = 0, 1, TILE'NO S "	PAGE DOTTE = PAGE NIOTH - 5 & PAGE PAGE PAGE PAGE PAGE PAGE PAGE PAGE	PAGE DYTE = PAGE NIOTH - 5 & PAGE PAGE PAGE PAGE PAGE PAGE PAGE PAGE	**************************************	PAGE ONTE = PAGE MIDTH - 5 8 PT PROR I = 0, 1, HEAD NO 8 PT PROR I = 0, 1, HEAD NO 8 PT PROR I = 0, 1, HEAD NO 8 PT PROR I = 0, 1, HEAD NO 8 PT PROR I = 0, 1, HEAD NO 8 PT PROR I = 0, 1, TITLE NO 8 PT PROR I = 0, 1, TITLE NO 8 PT PROR I = 0, 1, TITLE NO 8 PT PROR I = 0, 1, TITLE NO 8 PT PROR I = 0, 1, TITLE NO 8 PT PROR I = 0, 1, TITLE NO 8 PT PROR I = 0, 1, DITTE NO 8 PT PROR I = 0, 1, DIT	PAGE ONTE = PAGE MIDTH - 5 8 PAGE PAGE PAGE PAGE PAGE PAGE PAGE PAGE	FOR I = 0, 1, HEAD'NO S "  *** FOR I = 0, 1, HEAD'NO S "  *** FOR I = 0, 1, HEAD'NO S "  *** FOR I = 0, 1, HEAD'NO S "  *** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  ** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I =	PAGE DOTTE = PAGE NIOTH - 5 & PAGE PAGE PAGE PAGE PAGE PAGE PAGE PAGE	FOR I = 0, 1, HEADNO S "  *********************************	PAGE ONTE = PAGE NIOTH - 5 8 PAGE NIOTH - 5 8 PAGE NIOTH - 10
**************************************	*** *** *** *** *** *** *** *** *** **	**************************************	FOR I = 0, 1, HEAD'NO S "  *** FOR I = 0, 1, HEAD'NO S "  *** FOR I = 0, 1, HEAD'NO S "  *** FOR I = 0, 1, HEAD'NO S "  *** FOR I = 0, 1, TILE NO S "  *** I S "  *** I S "  *** I S "  *** I S "	PAGE DYTE = PAGE NIOTH - 5 & PAGE PAGE PAGE PAGE PAGE PAGE PAGE PAGE	FOR I = 0, 1, HEAD'NO S " FOR I = 0, 1, HEAD'NO S " FOR I = 0, 1, HEAD'NO S " FOR I = 0, 1, TILE'NO S "	PAGE DOTTE = PAGE NIOTH - 5 & PAGE PAGE PAGE PAGE PAGE PAGE PAGE PAGE	PAGE DYTE = PAGE NIOTH - 5 & PAGE PAGE PAGE PAGE PAGE PAGE PAGE PAGE	**************************************	PAGE ONTE = PAGE MIDTH - 5 8 PT PROR I = 0, 1, HEAD NO 8 PT PROR I = 0, 1, HEAD NO 8 PT PROR I = 0, 1, HEAD NO 8 PT PROR I = 0, 1, HEAD NO 8 PT PROR I = 0, 1, HEAD NO 8 PT PROR I = 0, 1, TITLE NO 8 PT PROR I = 0, 1, TITLE NO 8 PT PROR I = 0, 1, TITLE NO 8 PT PROR I = 0, 1, TITLE NO 8 PT PROR I = 0, 1, TITLE NO 8 PT PROR I = 0, 1, TITLE NO 8 PT PROR I = 0, 1, DITTE NO 8 PT PROR I = 0, 1, DIT	PAGE ONTE = PAGE MIDTH - 5 8 PAGE PAGE PAGE PAGE PAGE PAGE PAGE PAGE	FOR I = 0, 1, HEAD'NO S "  *** FOR I = 0, 1, HEAD'NO S "  *** FOR I = 0, 1, HEAD'NO S "  *** FOR I = 0, 1, HEAD'NO S "  *** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  ** FOR I = 0, 1, OUT** FOR I = 0, 1, TILE NO S "  *** FOR I =	PAGE DOTTE = PAGE NIOTH - 5 & PAGE PAGE PAGE PAGE PAGE PAGE PAGE PAGE	FOR I = 0, 1, HEADNO S "  *********************************	PAGE ONTE = PAGE NIOTH - 5 8 PAGE NIOTH - 5 8 PAGE NIOTH - 10
**************************************	*** **********************************	FOR I = 0, 1, HEAD'NO S "	FOR I = 0, 1, HEAD'NO S "  *********************************	FOR I = 0, 1, HEADNO S " FOR I = 0, 1, HEADNO S " FOR I = 0, 1, HEADNO S " FOR I = 0, 1, TILE"NO S " FOR I = 0, 1, TILE"	FOR I = 0, 1, HEAD'NO S " FOR I = 0, 1, HEAD'NO S " FOR I = 0, 1, HEAD'NO S " FOR I = 0, 1, TILE"NO S "	FOR I = 0, 1, HEADNO S " FOR I = 0, 1, HEADNO S " FOR I = 0, 1, HEADNO S " FOR I = 0, 1, TILE"NO S "	**************************************	FOR I = 0, 1, HEAD'NO \$ " FOR I = 0, 1, TILE'NO \$ " FOR I = 0, 1, TI	**************************************	*** *** *** *** *** *** *** *** *** **	FOR I = 0, 1, 0.01*BUF*SIZE - 1 ***  ***  ***  ***  ***  ***  ***	FOR I = 0, 1, HEADNO S " FOR I = 0, 1, HEADNO S " FOR I = 0, 1, HEADNO S " FOR I = 0, 1, TILE"NO S "	FOR I = 0, 1, HEAD'NO \$ "  *********************************	*** *** *** *** *** *** *** *** *** **
**************************************	FOR I = 0, 1, HEAD'NO \$ "  FOR I = 0, 1, TITLE NO \$ "  TITLE SH & "FOR I = 0, 1, TITLE NO 8 "	**************************************	FOR I = 0, 1, HEAD'NO S "  *********************************	FOR I = 0, 1, HEADNO \$ " FOR I = 0, 1, TILE NAME INDEX;);  **AMÉRITE BYTE = MAKGIN + **LENGTH (HEADER (SNAME INDEX;));  **ANTERNATE BYTE = MAKGIN + **ANTERNATE BYTE =	FOR I = 0, 1, HEAD'NO S " FOR I = 0, 1, TILE'NO S "	FOR I = 0, 1, HEADNO S " FOR I = 0, 1, HEADNO S " FOR I = 0, 1, HEADNO S " FOR I = 0, 1, TILE"NO S " FOR I = 0, 1	FOR I = 0, 1, HEAD'NO \$ " FOR I = 0, 1, TILE'NO \$ " FOR I = 0, 1, TI	FOR I = 0, 1, HEAD'NO S "	**************************************	**************************************	FOR I = 0, 1, 0.1 HEAD'S  ***********************************	FOR I = 0, 1, HEADNO S " FOR I = 0, 1, HEADNO S " FOR I = 0, 1, HEADNO S " FOR I = 0, 1, TILE"NO S "	**************************************	*** *** *** *** *** *** *** *** *** **
*** *** **** *************************	FOR I = 0, 1, HEAD'NO S ===================================	**************************************	TITLE SH SFOR I = 0, 1, TITLE NO SFOR I = 0, 1, TI	FOR I = 0, 1, HEAD'NO'S " FOR I = 0, 1, TILE'NO'S "	TITLE SH SFOR I = 0, 1, HEAD'NO S "  **STEER I = 0, 1, HEAD'NO S "  **STEER I = 0, 1, HEAD'NO S "  **STEER I = 0, 1, HEAD'NO S "  **TITLE SH S "FOR I = 0, 1, TITLE NO S "  **TITLE SH S "FOR I = 0, 1, TITLE NO S "  **TITLE SH S "FOR I = 0, 1, TITLE NO S "  **TITLE SH S "FOR I = 0, 1, TITLE NO S "  **TITLE SH S "FOR I = 0, 1, TITLE NO S "  **TITLE SH S "FOR I = 0, 1, TITLE NO S "  **TITLE SH S "FOR I = 0, 1, TITLE NO S "  **TITLE SH S "FOR I = 0, 1, TITLE NO S "  **TITLE SH S "FOR I = 0, 1, TITLE NO S "  **TITLE SH S "FOR I = 0, 1, TITLE NO S "  **TITLE SH S "FOR I = 0, 1, TITLE NO S "  **TITLE SH S "FOR I = 0, 1, TITLE NO S "  **TITLE SH S "FOR I = 0, 1, TITLE NO S "  **TITLE SH S "FOR I = 0, 1, TITLE SH	FOR I = 0, 1, HEAD'NO'S " FOR I = 0, 1, HEAD'NO'S " FOR I = 0, 1, HEAD'NO'S " FOR I = 0, 1, TILE'NO'S "	**************************************	##ANKON = MAGE NUOTH - S # PEGE BYTE = D. 1. HEAD'NO \$ "	**************************************	*** *** *** *** *** *** *** *** *** **	FOR I = 0, 1, HEAD'NU S "  ***FOR I = 0, 1, HEAD'NU S "  ****FOR I = 0, 1, HEAD'NU S "  *********************************	FOR I = 0, 1, HEAD'NO'S " SS) (HEADER(SS)) = 5H(PAGE) SNAME BYTE = MAKGIN + LENGTH(HEADER(SNAME'INDEXE)) + 1 S TITLE'SN S "FOR I = 0, 1, TITLE'NO'S " FOR I = 0, 1, TITLE'NO'S "	**************************************	*** **********************************
*** *** **** **** ********************	**************************************	##ADROND = HEAD NO	##ADROOM = HEAD'NO & 2 & PAGE'BYTE = PAGE'NDTH - 5 & PAGE'BYTE = PAGE'NDTH - 5 & PAGE'BYTE = D, 1, HEAD'NO & "  #################################	*** PAGE BYTE = PAGE *** PAGE ****  *******************************	##GARGON = HEAD'NO & C & FORE BYTE = D. 1. HEAD'NO & V = FOR I = D. 1. HEAD'NO & V = FOR I = D. 1. HEAD'NO & V = FOR I = D. 1. HEAD'NO & V = FOR I = D. 1. HEAD'NO & V = FOR I = D. 1. TITLE N	PAGE BYTE = PAGE NUDTH - 5 8  FOR I = 0, 1, HEAD'NO 8 *  FOR I = 0, 1, HEAD'NO 8 *  FOR I = 0, 1, HEAD'NO 8 *  FOR I = 0, 1, TILE'NO 8  TILE'SM 8 *FOR I = 0, 1, TILE'NO 8  FOR I = 0, 1, TILE'NO 8 *  FOR I = 0, 1, TILE'NO 8 *FOR I = 0, 1, TILE'NO 8 *  FOR I = 0, 1, OUT'BUF'SIZE - 1 *  FOR I = 0, 1, COUT'BUF'SIZE - 1 *	*** PAGE BYTE = PAGE ***  ********************************	##ADRONN = HEADEND NO	##ADRONN = HEADEND NO 2 & PAGE BYTE = PAGE NUDTH - 5 & PAGE BYTE = PAGE NUDTH - 5 & PAGE BYTE = D. 1, HEAD NO 8 **  ### ## ### ### ### ### ### ### ### ##	**************************************	##ADROOM = HEAD NO	PAGE BYTE = PAGE NUDTH - 5 & PAGE BYTE = PAGE NUDTH - 5 & PAGE BYTE = D, 1, HEAD'NO \$ **  ***PAGE BYTE = D, 1, HEAD'NO \$ **  ****PAGE BYTE = PAGE NUDTH - 10, 1, S\$ (HEADER(\$S)) = 5H(PAGE);  ***********************************	##ADRONN = HEADNON = LEGE   PAGE   PA	##ADRONG = HEADNON = HEADN
**************************************	**************************************	**************************************	*** **********************************	*** **********************************	** ** ** ** ** ** ** ** ** ** ** ** **	*** **********************************	*** **********************************	*** *** *** *** *** *** *** *** *** **	*** *** **** **** ********************	*** **********************************	** ** ** ** ** ** ** ** ** ** ** ** **	*** **********************************	*** *** *** *** *** *** *** *** *** **	*** **********************************
######################################	**************************************	**************************************	**************************************	**************************************	*** **********************************	**************************************	**************************************	*** *** *** *** *** *** *** *** *** **	*** **********************************	*** **********************************	*** *** *** *** *** *** *** *** *** **	*** **********************************	*** *** *** *** *** *** *** *** *** **	*** **********************************
**************************************	**************************************	**************************************	**************************************	**************************************	*** *** *** *** *** *** *** *** *** **	**************************************	**************************************	*** *** *** *** *** *** *** *** *** **	*** *** *** *** *** *** *** *** *** **	**************************************	**************************************	**************************************	*** *** *** *** *** *** *** *** *** **	**************************************
*** **********************************	*** **********************************	PAGE BYTE = PAGE WIDTH - 5 %  PAGE BYTE = PAGE WIDTH - 5 %  PFOR I = 0, 1, HEAD NO %  BYTE (\$PAGE NIDTH - 10, \$ %  NAME BYTE = MAKGIN + 10, \$ %  LENGTH (HEADER(SNAME 'INDEXS))  + 1	*** **********************************	*** **** *****************************	PAGE BYTE = PAGE WIDTH - 5 %  PAGE BYTE = PAGE WIDTH - 5 %  FOR I = 0, 1, HEAD NO %  BYTE (\$PAGE WIDTH - 10,  S\$) (HEADER(\$G\$)) = 5H(PAGE)  ***  ***  ***  ***  ***  ***  ***	*** **********************************	*** **********************************	*** **** *****************************	*** **********************************	PAGE BYTE = PAGE WIDTH - 5 %  PAGE BYTE = PAGE WIDTH - 5 %  PAGE BYTE = D. 1, HEAD NO & "  BYTE (SPAGE WIDTH - 10, %  SS) (HEADER(SS)) = 5H(PAGE)  NAME BYTE = MAKGIN +  LENGTH(HEADER(SNAME INDEXS))  1	PAGE BYTE = PAGE WIDTH - 5 %  PAGE BYTE = PAGE WIDTH - 5 %  POR I = 0 . 1 HEAD NO %  BYTE (\$PAGE WIDTH - 10 .  5 \$) (HEADER(\$\$)) = 5H(PAGE )  **ARE BYTE = HARGIN +  **LENCTH(HEADER(\$NAME 'INDEX\$))  ** 1 %  ***ARE BYTE = HARGIN +  ***LENCTH(HEADER(\$NAME 'INDEX\$))  ***ARE BYTE = HARGIN +  ***LENCTH(HEADER(\$NAME 'INDEX\$))  ***ARE BYTE = HARGIN +  ***ARE BYTE = HARGIN	*** **********************************	*** **********************************	PAGE BYTE = PAGE WIDTH - 5 %  PAGE BYTE = PAGE WIDTH - 5 %  POR I = 0, 1, HEAD NO & "  BYTE (\$PAGE WIDTH - 10,  5 \$) (HEADER(\$\$)) = 5H(PAGE)  **ARE BYTE = MARGIN +  **LENGTH(HEADER(\$NAME 'INDEX\$))  ** LENGTH(HEADER(\$NAME 'INDEX\$))  ** LA \$  ***ARE BYTE = MARGIN +  ***LENGTH(HEADER(\$NAME 'INDEX\$))  ***ARE BYTE = MARGIN +  ***ARE BYTE = MA
PAGE BYTE = PAGE NO + 2 & PAGE BYTE = PAGE NIDTH - 5 & PAGE NIDTH - 10 + 10 + 10 + 10 + 10 + 10 + 10 + 10	PAGE BYTE = PAGE NIDTH - 5 %  PAGE NITH - 5 %  POR I = 0, 1, HEAD NO %  POR I = 0, 1, HEAD NO %  POR I = 0, 1, HEAD NO %  POR I = 0, 1, TILE NO %	PAGE BYTE = PAGE NIDTH - 5 %  PAGE BYTE = PAGE NIDTH - 5 %  POR I = 0, 1, HEAD NO & "  BYTE (\$PAGE NIDTH - 10, 5 %)  SA) (HEADER(\$G\$)) = 5H(PAGE)  NAKE BYTE = HAKGIN + LOGTH (HEADER(\$NAME INDEX\$))  LENGTH (HEADER(\$NAME INDEX\$))  LA S	PAGE BYTE = PAGE NO + 2 & PAGE	PAGE BYTE = PAGE NIDTH - 5 %  PAGE BYTE = PAGE NIDTH - 5 %  BYTE (\$PAGE MIDTH - 10, 5 %)  SANARE BYTE = MAKGIN + 10, 5 %  LENGTH (HEADER(SNAME INDEXS))  + 1 %  - 1 %  - 1 %  - 2 %  - 3 %  - 4 %  - 4 %  - 4 %  - 4 %  - 5 %  - 6 %  - 6 %  - 7 %  - 7 %  - 7 %  - 7 %  - 7 %  - 8 %  - 9 %  - 1 %  - 2 %  - 3 %  - 4 %  - 4 %  - 4 %  - 5 %  - 6 %  - 7	PAGE BYTE = PAGE NO + 2 & PAGE BYTE = PAGE NO + 2 & PAGE BYTE = PAGE NOTH - 5 & PAGE NOTH - 10 + 10 + 10 + 10 + 10 + 10 + 10 + 10	PAGE BYTE = PAGE NO + 2 & PAGE BYTE = PAGE NO + 2 & PAGE BYTE = PAGE NOTH - 5 & PAGE NOTH - 10 + 10 + 10 + 10 + 10 + 10 + 10 + 10	PAGE BYTE = PAGE NIDTH - 5 %  PAGE BYTE = PAGE NIDTH - 5 %  PYOR I = 0, 1, HEAD NO & "  BYTE (\$PAGE NIDTH - 10, 5 %)  SAN (HEADER(\$\$)) = 5H(PAGE)  NAME BYTE = MAKGIN + 10, 5 %  LENGTH(HEADER(\$NAME INDEX\$))  + 1 %  - 1	PAGE BYTE = PAGE NIDTH - 5 %  PAGE BYTE = PAGE NIDTH - 5 %  POR I = 0, 1, HEAD NO & PAGE NIDTH - 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,	PAGE BYTE = PAGE NIDTH - 5 %  PAGE NIDTH - 10 %  PAGE	PAGE BYTE = PAGE NIDTH - 5 %  PAGE NIDTH - 5 %  PAGE NIDTH - 5 %  PAGE NIDTH - 10 %  PAGE	PAGE BYTE = PAGE NIDTH - 5 %  PPAGE BYTE = PAGE NIDTH - 5 %  FOR I = 0, 1, HEAD NO %  BYTE (SPAGE NIDTH - 10, 6 %  S\$) (HEADER(SS)) = 5H(PAGE)  LENGTH (HEADER(SNAME 'INDEX\$))  * LENGTH (HEADER(SNAME 'INDEX\$))  * 1 %  * NASE OF STATE = NASE OF STATE OF STA	PAGE BYTE = PAGE NO + 2 & PAGE BYTE = PAGE NO + 2 & PAGE BYTE = PAGE NO TH - 5 & PAGE NO TH - 10 + 10 + 10 + 10 + 10 + 10 + 10 + 10	PAGE BYTE = PAGE NIDTH - 5 %  PAGE BYTE = PAGE NIDTH - 5 %  POR I = 0, 1, HEAD NO & PAGE NIDTH - 5 %  BYTE (SPAGE NIDTH - 10, 5 %)  SS) (HEADER(SS)) = 5H(PAGE)  NAKE BYTE = MAKGIN + LEGTH(HEADER(SNAHE INDEXS))  LENGTH(HEADER(SNAHE INDEXS))  1	PAGE BYTE = PAGE NIDTH - 5 %  PAGE NIDTH - 5 %  PAGE NIDTH - 5 %  POR I = 0, 1, HEAD NO %  BYTE (\$PAGE NIDTH - 10, 5 %)  BYTE (\$PAGE NIDTH - 10, 5 %)  PAGE NIDTH - 10, 5 %  BYTE E HARGIN + 10, 5 %  LENGTH (HEADER(\$NAME 'INDEX\$))  ***  ***  ***  ***  ***  ***  ***
PAGE BYTE = PAGE NIDTH - 5 & PAGE BYTE = PAGE NIDTH - 5 & PAGE BYTE = PAGE NIDTH - 5 & PAGE BYTE = PAGE NIDTH - 10 & PAGE BYTE	PAGE BYTE = PAGE NIDTH - 5 & PAGE BYTE = PAGE NIDTH - 5 & PAGE BYTE = PAGE NIDTH - 5 & PAGE BYTE = PAGE NIDTH - 10 & PAGE BYTE SPAGE NIDTH - 10 & PAGE BYTE = PARGIN + 10 & PAGE BYTE = PAGE BYTE = PARGIN + 10 & PAGE BYTE = PAGE B	PAGE BYTE = PAGE NIDTH - S & PAGE BYTE = PAGE NIDTH - S & PAGE BYTE = PAGE NIDTH - S & PAGE	FOR I = 0, 1, HEAD'NO \$ " FOR I = 0, 1, TILE'NO \$ " FOR I = 0, 1, TI	HEADENON = HEADENO + 2 & PAGE BYTE = PAGE WIDTH - 5 & PAGE BYTE = PAGE WIDTH - 5 & PAGE WIDTH - 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10	FOR I = 0, 1, HEAD'NO + 2 & PAGE'BYTE = PAGE'NIDTH - 5 & PAGE'BYTE = PAGE'NIDTH - 10, 1, HEAD'NO & PAGE'S (REA) = 54(PAGE) = 55(PAGE) = 54(PAGE) = 54(PAGE	** ** ** ** ** ** ** ** ** ** ** ** **	PAGE BYTE = PAGE WIDTH - S & PAGE BYTE = PAGE WIDTH - S & PAGE BYTE = PAGE WIDTH - S & PAGE BYTE = PAGE WIDTH - 10.  SYTE (SPAGE WIDTH - 10.  SANKE BYTE = PAGE NOES NOES S & PAGE BYTE = PAGE S & PAGE S	HEADENON = HEADENO + 2 s PAGE BYTE = PAGE NIDTH - 5 s PAGE BYTE = PAGE NIDTH - 10 s	PAGE BYTE = PAGE NIDTH - 5 & PAGE BYTE = PAGE NIDTH - 5 & PAGE BYTE = PAGE NIDTH - 10	HEADENON = HEADENO + 2 & PAGE BYTE = PAGE NIDTH - 5 & PAGE BYTE = PAGE NIDTH - 10 & 1 & HEADENO & PAGE BYTE E HARGIN + 10 & 5 & 10 & 10 & 10 & 10 & 10 & 10 &	PAGE BYTE = PAGE NIDTH - 5 & PAGE BYTE = PAGE NIDTH - 5 & PAGE BYTE = PAGE NIDTH - 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10	** ** ** ** ** ** ** ** ** ** ** ** **	PAGE BYTE = PAGE NIDTH - S & PAGE BYTE = PAGE NIDTH - S & PAGE BYTE = PAGE NIDTH - S & PAGE BYTE = PAGE NIDTH - 10 & PAGE PAGE PAGE NIDTH - 10 & PAGE PAGE PAGE PAGE PAGE PAGE PAGE PAGE	PAGE BYTE = PAGE NIDTH - 5 & PAGE BYTE = PAGE NIDTH - 5 & PAGE BYTE = PAGE NIDTH - 10    SYPE(SPAGE NIDH - 10    SYPE(SPAGE NIDTH - 10    SYPE(SPA
**************************************	*** *** **** *************************	*** **********************************	**************************************	** ** ** ** ** ** ** ** ** ** ** ** **	**HEADENOOM = 16 S ** **HEADENOOM = 16 S ** **PAGE**** **PAGE*** **PAGE** **PAGE*** **PAGE**	*** **********************************	*** **********************************	*** **********************************	*** **********************************	*** **********************************	**************************************	*** **********************************	*** *** **** *************************	*** **********************************
**************************************	**************************************	**************************************	**************************************	##EADROOM = 16	**************************************	*** *** *** *** *** *** *** *** *** **	##EADROOM = 16	**************************************	##EADROOM = 168 ##EADROOM = 168 ##EADROOM = 168 ##EADROOM = 162 ##EADROOM = 168 ##EADROOM = 16	*** *** **** **** ********************	**************************************	*** *** *** *** *** *** *** *** *** **	**************************************	**************************************
HEADING & ***********************************	**************************************	**************************************	######################################	##ADECOM = HEADONO 4 2 8  ***PAGE*********************************	###ADTSPLAY************************************	###ADTONOM = ##ADD NO # 2 # 2 # 2 # 2 # 2 # 2 # 2 # 2 # 2 #	##ADENDOM = HEADONO & 2 & PAGE BYTE = PAGE WIDTH - 5 & PAGE BYTE = PAGE WIDTH - 5 & PAGE BYTE = 0, 1, HEADONO & PAGE BYTE = MAGIN - 10, 1 & HEADONO & PAGE BYTE = MAGIN - 10, 1 & HEADONO & PAGE BYTE = MAGIN - 10, 1 & PAGE BYTE	##ADROOM = HEADNO & 2 & PAGE "NIDTH - 5 & PAGE "NIDTH - 10 & PAGE "NIDTH -	**************************************	### ### ### ### ### ### ### ### ### ##	###ADROOM = HEAD*NO 2 & PAGE*BYTE = PAGE*B	**************************************	**************************************	######################################

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER DESIGN DIAGRAM OF PART2"INIT

\*\*\*\*\*\*\*\*\*\*

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of Phzerr

ERROR HESSAGES AND STOPS ERECUTION OF THE DOG. " EXECUTION OF THE DOG. "  THE ERROR HESS CHARACTER B "  THE ERROR HESS CHARACTER B "  OUT (ERROR HESS SH) B "	ERROR MESSAGES AND STOPS ERECUTION OF THE DDG. " EXECUTION OF THE DDG. "  THE ERROR MESS CHARACTER S. "  THE ERROR MESS SHI S. "  OUTICEROR WESS. SHI S. "
IF DE BUG27 \$	WRITE BLOCKS TO DISK
	FEETH **NEW*TLE3 & **HESSAGE = 6H(FILE31) & ***********************************
Market of the state of the stat	**************************************
\$ 401°	1

\*PROC PLACE (NEW BOX) \$

\*\* THE PROCEDURE PLACE PLACES
THE CURRENT CODE BLOCK ON THE
DESIGN DIAGRAM. IT ALSO
PERFORMS STUMP DETECTION. \*\* \*

\*\* PAGE ON WHICH THE BOX BOTTOM\*
\*\* APPEARS\*\*
\*\* ITEM ENTRANCE INTEGER \$
\*\* "VERTICAL OR HORIZONTAL ENTRY\*
\*\* INTO BLOCK\*\*
\*\* ITEM FATHER\*BOTTOM INTEGER \$
\*\* "BOTTOM LINE OF FATHER CODE \*\*
\*\* BLOCK\*\* \*ITEM FATHER LEFT INTEGER \$

\*\*\*FATHER'S STARTING COLUMN\*\*

\*ITEM FATHER RIGHT INTEGER \$

\*\*\* FATHER'S ENDING COLUMN\*\*

\*ITEM FATHER INTEGER \$

\*\*\*FATHER'S FILES RECORD

\* NUMBER\*\*

\*ITEM HID INTEGER \$

\*\*\* PAGE ON HHICH BLOCK'S

\* HIDPOINT APPEARS\*\*

\*ITEM NEW\*BOX INTEGER;

\*\*\*NEM FS REC DESCRIBING LATEST\*

\*\*\*BOX\*\*\* \*ITEM PAGE SPANS INTEGER \$
\*\*\*NUMBER OF PAGES SPANNED BY
\* BLOCK\*\*

\*ITEM STUMP\*REF\*BOTTOM INTEGER \* \$ \*\*\* BOTTOH OF STUNP REF \* DISPLAY\*\*
\*ITEH TOP INTEGER \$

\*ITEM REMAINTEGER \$

\*ITEM REMAINTEGER \$

\*\*\*PAGE DISPLACEMENT\*\*

\*ITEM REMAINTEGER \$

\*ITEM REMAINTEGER \$

\*\*\*PAGE DISPLACEMENT\*\*

\*ITEM STMT\*\*TYPE INTEGER \$

\*\*\*TYPE OF CURRENT FILES REC\*\*

```
*HIDT4 = MIN(WIDTH, MAX*WIDTH) *
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    ***TRUNCATE IF NECESSARY **
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 *****************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                . LINES($ ACCESS3 (NEW BOX) $1 2"
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       ******************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  *******************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           *NIDT4 = MIDIH + 2 +
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
Design diagram of place
                                                                                                                                                                                    **--*HESSAGE = 6H(PLACE ) $

*T*HESS = IFORMAT(NEW*BOX) $

*HESSAGE = CAT(HESSAGE, T*HESS)*
                                                                                                                                                                                                                                           *OUT(MESSAGE, MESS'SW) $
                                                                                                                                                                        ************************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             ******************
                                           * APPEARS**
* ITEM VERT INTEGER P 1 $
****CONSTANT INDICATING VERTICAL*
                                                                                                                                                                                                                                                                                                * ENTRY**
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     *WIDTH = BLOCK *WIDTH(SACCESS3(N*
                                                                                                                                                                                                                                                                                     ***********************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                         *********************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     *Eh-BOX)S) S
                                                                                                                                                                                                                                                                                                                                                                                                               . .. DETERMINE THE ACTUAL BOX .. WIDTH ..
                                                                                                                                                                      *IF 0E BUG20 $ ...
                                                                                                                                                                                                                *************
                                                                                                                                                          *************
```

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF PLACE

PIF STHITTYPE EQ CONTROL"S & "	**************************************		** BOX 4AS A FATHER IT'S NOT A PROC.CLOSE OR PROGRAM HEAD**	### ##################################	**************************************	**************************************	H *****FATHER GR G S ****** WE ARE NOT BEGINNING A STUMP ** STUMP **
		 a S C + HIOIN   HIOIN	VIFEITH *STMTTTYPE NO CONTROL'S	TELITH.	Self-cable of States, 1975, 18	CONTRACTOR OF THE PROPERTY OF	STATE OF THE STATE

C S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of place

.. IS NEW BOX A STUMP ..

			ecsally, Thursday 105 200 serva, box alex						A STUMP		UND = TRJE S .		*************************								SPAN A PAGE			F CAN CF SC 2 (NEUF	GTH = TOP. •	•	CACCESSS (NEM**	•	***************************************		:		*1S) + HEADROOM - RENI S	# STOP LINE (SACCESSS (NEW BOX)S)	** + HEADROOM - REM: S		***************************************
	STOP COL (SACCESSINE W BOX) \$) =+	START COL (SACCESS3(NEW-BOX)S)+		THE CONTRACTOR OF THE PROPERTY		***************************************	*IF STOP*COL (SACCESS3 (NEW* BOX) \$ *	:	V S.11	• *************************************	STUMP FOUND	*RETURN \$			STOP LINE (SACCESSA(NEW-BOX)S) .	= START"LINE (SACCESS3(NEW BOX+	+ LINES (SACCESSS (NEW BOX)S+	***************************************		•	********* DOES NEW BOX SPAN A PAGE			#DEMOCRATION OF THE CANADA	**BOX) \$ ) , PAGE*LNGTH = TOP	* RENI) \$	**ROUD(STOP*LINE (\$ACCESS3 (NEW**	* REH2) \$	**********************	******	*IF REM1 LQ HEADROOM \$	*************		•		•	*************
***************************************	STOP COL (SACCI	START COLISA	T - HIGH - T	WALLES - INUE S	•	************	*IF STOP*COLIS	.) GR PAGE . MEDTH			•	•		*************	*STOP LINE (SAC	. = START*LINE	15) + LINES(\$4	***************************************	•	+		•	• • •	•			•	•	• •		•			•		•	

2)

RETURN S .

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of place	FROM 120 ·	PRESENTSIMITYPE EQ COMMENTOZ 8START-LINE (SACCESSSINEN-BOX) 8START-LINE (SACCESSSINEN-BOX) 8START-LINE (SACCESSSINEN-BOX) 8	* ************************************	ALTERIAL OF A STATE TO CONTROL W WAS TO COLLOSS (NEW-BOX) STATE OF THE PROPERTY OF THE PROPERT	ORIF 1 SSTART-COL(SACE SSJ(ME W BOX) S) +ORIF 1 SSTART-COL(SACE SSJ(ME W BOX) S) +
C S DRAPER DESIGN DIAG	. 124 FROM 120 ·	 IFEITH .		· IFEITH	

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of place

. 125 FROM 120 \*

•	
*******	
HITTH	F FFATHER TYPE EQ CONTROL 3 AND 4 correspondences consesses
********	* STHT'TYPE NO CONTROL'S S ***START'COL (SACCESSSINEM'BOX)SI*
	•
	OF THE FATHER. IYPE NO CONTROL
	在中华的自己的国际中的一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个一个
	+-*ORIF 1 \$ *\$1ART*COL(\$ACCESS3(NEW*BOX)\$) \$
	********* * FAMER-LEFT &
********	
IFEITH	1
	* ********************* * = MAX(LAST *LINE + 2, *
	***************************************

	"IT CAN"T BE AVOIDED ""	ARENQUO (NIDPOINT, PAGE "LNGTH as the transfer of the transfer			MOVE	0X) \$) *	(\$ £) * * 100 x * 30 x 1 x *	* * * * * * * * * * * * * * * * * * *		
**TRY NOT TO LET A CONTROL BOX SPAN A HEADING **	**************************************	RENQU	129*	**************************************	BOX TO NEXT PAGE	*START'LINE(\$ACCESSINEW*BOX) * * STOP'LINE(\$ACCESSINEW*BOX) *	*STOP*LINE(\$ACCESSINEW*BOX)\$) * * START*LINE(\$ACCESSINEW*BOX)\$) * * START*LINE(\$ACCESSINEW*BOX)\$) * *) * LINES(\$ACCESSINEW*BOX)\$*	*HIDPOINT = (START-LINE(\$ACCESS* *3(NEM*BOX)\$) + STOP*LINE(\$ACCE* *\$\$3(NEM*BOX)\$) / Z * *********************************	•••	* * * * * * * * * * * * * * * * * * *
PE GG CONTROL'1 S+ STAT	5 I 5 I 5 I 5 I 5 I 5 I 5 I 5 I 5 I 5 I						5 % 0 # -0 % -0 % -0 % -0 % -0 % -0 % -0 % -0 %			
ATTENTA	•••••	••••			**************************************		* * * * * * * * * * * * * * * * * * *	100 100 100 100 100 100 100 100 100 100	A STATE OF THE STA	を 日の の 1 年 日の 1 日の 1 日の 1 日の 1 日の 1 日の 1

. . NOT A CONTROL BOX ..

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of place

********************	_	. * STOP LINE (SACCESS3 (NEW BOX) *	Ī	•
:	STOP LINE (SACCE SS3(NEW BOX)S)	8		* PAGE SPANS \$
•	80	×		
:	E K	35	3	
:	Z	SS	2	
:	SS	3		
:	300	3		
:	*	NE	RO	2
:	ME	7	EAD	PAH
:	7	90	Ξ	
:	9	S	*	AGE
:	ST	*	** + THEADROOM - 11 .	. PAGE SPANS &
•	•	•	•	

C S DRAFER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design Diagram of Place	IAGRAMER
***************************************	
* 129 FROM 127 *	
*************	
*IF TOP NQ MID \$ *****	
INIODOIN ISHTOW	
•	
******************	*****
* MIDPOINT = MIDPOINT + MEADONNA	******
* * (MID - TOP) \$	
***********	*****
***************************************	
*STOP*LINE(SACCESS3(NEW-BOX)S) *	
. = STOP*LINE(SACCESS3(NEW*BOX)*	
*\$) + (HEADROOM - 1) *	
* (PAGE SPANS) \$	

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER

TAGKAMIEK			HOVE BOX BACK 1 COLUMN	
			BACK	
3			80X	
0 2 2			HOVE	
			:	
DESIGN DIAGRAM OF PLACE	* 130 FROM 127 *	••		

C-S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DEMAGN DIAGRAM OF POP-LAYOUT-INFO

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* THE PROCEDURE POP\*LAYOUT'INFO POPS THE TOP ELEMENT OF LAYOUT'STACK INTO LAST'LINE, \*\* \*PROC POP-LAYOUT INFO \$ \*

\*LAST\*LINE = LAYOUT\*STACK(\$LAYO\*
\*USTACK\*TOP\$) \$
\*CUR\*GROUP = GROUP\*STACK(\$LAYOU\*
\*T\*STACK\*TOP\$) \$
\*LAYOUT\*STACK\*TOP = \*
\*LAYOUT\*STACK\*TOP = \*
\*RETURN \$

\*ITEM POP\*RC INTEGER S \*

RETURN S

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANMER DESIGN DIAGRAM OF PUSH"LAYOUT"INFO

** THE PROCEDURE PUSH-LAYOUTINED STACK TOP OF LAYOUTSTACK TOP GR * ** ** ** ** ** ** ** ** ** ** ** **	SIGN DIAGRAM OF PUSH"LAYOUT"INFO
PROCEDURE  LAYOUT IND PUSHES THE  OF LAST LINE ONTO THE  LAYOUT STACK TOP GR **  **LAYOUT STACK	ROC PUSH'LAYOUT'INFO & .
LAYOUT STACK HAX &	D PUSHES INE ONTO
	*LAYOUT-STACK-TOP GR *

* S * S * PUSH* PROC* STACK (PUSH* REC) *
TWF PROCEDURE
. 02
IRRENT F
* RECORD ONTO THE PROC'STACK.
PITEM PUSH REC INTEGER \$
SASSESSES OF THE RECORDS OF
***************************************
wif DE BUG21 8 . sendentstatestatestatestates
8 ( HSNA) 8
. TOMESS . IF ORMAT (PUSH'R
STATEMENT OF THE CATCHESSAGE TO MESS OF
* S (NS.SSTEE NESSTEE . MESS.SR) &
. THIS PROC IS NOT NECESSARY
SUPPORTING NEXTED PROCS
* MOULD REQUIRE IT. **
TO THE PROPERTY OF THE PROPERT
. OVERFLOW) S
· sessessessessessessessessessesses
***************************************
PROC'STACK-TOP + 1 S
\$ \$

\*IF PROC'STACK\*TOP GR I \$ \*--\*PROC'FLAG = TRUE \$ \*

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design Diagram of Push-Prog-Stack

\*RETURN \$ \*

RESOLVE-STUMP FINDS THE ROOT
OF A STUMP AND INVOKES PLACE
FOR EACH RECORD CURRENTLY
WANGING OFF THE STUMP ROOT. GROUP TABLE.

GROUP TABLE.

GROUP TABLE.

CAUSING STUMP.

GROUP.

GROUP.

TIEM LAST INDEX INTEGER 8

TIEM LAST INDEX INTEGER 8

TIEM CLD. INDEX INTEGER 8

TIEM CLD. INDEX INTEGER 8

TIEM STUMP. RC INTEGER 8

TIEM SUB"STUMP INTEGER 8 \*ITEM DISPLAY\*ROOM B \$ \*\*\* ENDUGH ROOM FOR STUMP REF \*\* DISPLAY \*\* \* \* "" COMPLETION FLAG""
"ITEM FATHER INTEGER &
""RECORD NUMBER OF FATHER""
"ITEM HORIZ INTEGER P 0 \$ .PROC RESOLVE STUMP (STUMP REC) \*ITEM TYPE INTEGER P 0 \$
\*\*\* HORIZ-VERT FLAG\*\*
\*\*\* CONSTANT INDICATING VERT TF DE BUGZ6 \$ .

\*\*\*\*\*\* RESSAGE = 16H(STUMP CAUSE REC \* .T. HESS = IF ORMAT (STUMP. REC) & .

.............

\*

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER. DESIGN DIAGRAM OF RESOLVE'STUMP

** SAGE = CAT(MESSAGE, T** MESS)**  * S ** **OUT(MESSAGE, MESS*SW)	THAT STUMP IS HAT ITS FATHER MAS MP PAGE REFERENCE	**************************************	TIF STUMP REC LQ 0 8 **********************************	** ROOM FOR DISPLAY BOX ** **********************************
·ÆSS)•			TE STUMP REC LQ 0	

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DÍAGRAMER Design diagram of Resolve"Stump

STUR STUR
**************************************

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER DESIGN DIAGRAM OF RESOLVE. STUMP

C S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIACRAMER Design Diagram of Resolve-Stunp

* TUMPRECIST (STUMP REC)\$) * *********************************	. "FIND THE SUB"S ENTRY . IN GROUP".	* SUB"STUMP AND INDEX GR &] * *********************************	**************************************	** REORDER THE STUMPS**  **NEXT(SLAST*INDEXS) =	TELLASTORES COLD'INEX S CLASTORES COLD'INEX S CLASTORES COLD'INEX S CLASTORES COLD'INEX S C	*IF LAST*STUMP EQ OLO*INDEX \$ **LAST*STUMP = INDEX \$ * **********************************

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of Resolve-Stump

**************************************	A S S S S S S S S S S S S S S S S S S S	**REORDER THE STUMPS**  ***REORDER THE STUMPS**  ***NEXT(SLAST*INDEXS) =
	LS OS STUMP REGIST OF STATE OF	A CONTRACTOR OF THE CONTRACTOR
		The black of the state of the s

**************************************	**************************************	+-+ORIF 1 \$IF DEBUG26 \$HESSAGE = LTH(RESOLVEITREE END-  + 1 HESS = LFORMAT(STUMP-REC) \$ +  + 1 HESS = LFORMAT(STUMP-REC) \$ +  + 1 HESS = LFORMAT(STUMP-REC) \$ +  + 1 HESSAGE = CAT(MESSAGE, T-HESS) +  + 2 HESSAGE, HESS-SM) \$ +  + 3 HESSAGE, HESS-SM) \$ +  + 4 HESSAGE, HESSAGE, HESSAGE, HESS-SM) \$ +  + 4 HESSAGE, HESSAGE, HESSAGE, HESS-SMORTH \$ +  + 4 HESSAGE, HESSAGE, HESSAGE, HESS-SMORTH \$ +  + 4 HESSAGE,	SUB-TREE. LOOK FOR LEFT SUB-TREE.  SUB-TREE.  TO TRAVERSE	**PONTALE CLYPTRISACCESSISTUM * **PONTALE CLYPTRISACCESSISTUM * ***PONTALE CLYPTRISACCESSISTUM * ***DON STUMP'REC EQ **********************************
	. HE I I I I I I I I I I I I I I I I I I			

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design Diagram of Resolve-Stump

**************************************		# # # # # # # # # # # # # # # # # # #	

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER

1144 FROM 138 \*\*

1145 FROM 138 FR

G S ORAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAN OF RESOLVE-STUMP

\* 145 FROM 138 \*\*

\*\*YPE = VERT \$

\*\*FATHER = BACK\*V(\$ACCESS3(STUMP\*

\*\*FATHER = TRUE \$

\*\*SFLAY\*ROOM = TRUE \$

\*\*

146 FROM 140 ·

PLAST INDEX = INDEX S PINDEX S PAINDEX S PAIND

•								
					:		•	:
DESTON DIAGRAM OF RESULVE STO					********************		*INDEX = NEXT (SINDEXS) \$ *	******************
5					•		=	•
ÿ					i	-	×	•
					ï	2	8	:
5					:	2	Z	:
	:	-:			:	-	:	:
	:	3			:		=	:
9	•				•	×	ē	•
3	•	0			•	Ö	-	•
-	•	.:			÷	=	"	÷
5	:	.:			:	:	EX	:
7	:	4:			:	AS	2	:
K	***************************************	. 147 FROM 141 *	•	•	:	-	-	:

	:	OUTPUTS		THE		::	•	•	•	•		•		•	•		***	::	::		. 48		•	•
*PROC STARS(COL, LNGTH, LINE)		9	IRS	I COLUMN COL OF	:	********		WHICH STARS			NH ICH					:	:	:	:	:	NE (\$/	283		
•		ARS	ST	3	ORO.	:		H S			IN			OUTPUT		:	:	:	:	:	17.1	BYTE		
15		2	GTH	Z S	REC	:		HIC			1 0			5		:	:	:	:	:	1) (OUT-			
3		JUR.	2	STARTING ON COLU	5	-	8			ER	RECORD		NTEGER	ARS	4 2	:	:	:	:	:	÷	-		
COL		DCE	8	S	50	*******	COL INTEGER	Z		INTEGER		3	_	ST	M	:	:	:	:	:	LNGT	CHE	3	
RSC		8	FIZ	ING	=	:	H	20			PUTOUT	PPE		6	œ	:	:	:	:	•	:	SS-OUT (LIN	(ST	
STA		THE	ST	ART	¥	:	60	EC	:	LINE		Y S	ž	0	STA	٤	:	:			800	3	E	×
9		:	•	S	3	:	EH	•	TAR	EH	TE	TAR	EH	M	EH	324	:	:	:			•	NGT	TURN
·	• • •		•	•		:	11.	:		11.	:		11.	:	11.	-	:	:	:	:	18.	3		·RE

G S DRAFER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANMER DESIGN DIAGRAM OF TRANSFER\*WRITE3

	TATELIAN CONTRACTOR OF THE PROPERTY OF THE PRO	- certetees certeteesteesteesteesteesteesteesteestees				ATTEITH P-4-01 EQ F3.BLK & P-4-00UTPUT FILE3°2 REGS'IN'BLK3,0 COCCUPAT FILE3°2 REGS'IN'BLK3,0	**************************************	
FILE3'1 " FILE3'1 " FILE3'1 " FILE3'1 " FOR NAPUT FILE3'2 S *	FOR E = 0, 1, F3 EMPIY - 1 S		*SMUT INPUT FILE3*2 8 * *SMUT OUTPUT FILE3*1 8 *	. TRANSFER FROM FILE3"1 TO	*OPEN IMPUT FILE3*1 & * *OPEN OUTPUT FILE3*2 & *	This is a second		SHUT INPUT FILE3°1 S *
THE	10 A C 10				STATE State ST & CERTS	THE CONTROL OF STATE		DOMEST LANGUES ON THE WAY OF STREET

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANMER DESIGN DIAGRAN OF TRANSFER WRITES

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DESIGN DIAGRAM OF TRANSFER HRIT 153 FROM 151 \*\*

OUTPUT FILE 3'1 RECS'IN'BLK3,\*

F 3'8 UF 8

OR FROM BANCHER OF SERVICES THEORY OF SERVICE OF THE SERVICE OF TH

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of transfer writer

SAME AS TRANSFER MITE 3			
ONLY IT OPERATES ON FILE 4 BUFFER			
TABLE TMP'BUF R F4'BLKSIZ S N°  0 S  0 S  1 IEM ENT INTEGER S  1 IEM ENT INTEGER S			
PITEM ENT3 INTEGER 8  PEND  PITEM TMPI INTEGER 8  PECS INVOLKE FOR		**************************************	The state of the s
TF DE BUGS 8 . THE SSAGE . THE	**************************************		
F. SLK + 11 S C C C C C C C C C C C C C C C C C	Efficación de effectiva de effetiva de eff		
		2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	45.74 TO
La des estadas	***************************************		

A STREET OF STREET	NEWS STROOM WO NEWS STROOM WAS CONTINUED WAS	ecception of the control of the cont
SMUT IMPUT FILES * * * * * * * * * * * * * * * * * * *	INPUT FILEY 2 & CUIPUT FILEY 2 & C	
	** TRANSFER FROM FILE**1 TO FILE**2 **	
**************************************	* * * * * * * * * * * * * * * * * * *	
	L'EMPTY - 1 8 C1FELTH C-1-	STOREST DO 1. F4 EMPTY - 1.8 4
		++ORIF 1 8 *POS(FILE4*1) = 1 S ++ORIF 1 8 *POS(FILE4*1) = 1 S ++ORIF 1 8 *POS(FILE4*1 TMP1, TMP*8UF S *
*SAUT DUPUT FILE+*:		

w	
~	
5	
-	
C	
3	•
~ "	
n	•
- :	
3:	
-	
2 .	
= 2	
> 2	
2 0	
ŌF	
K O	
0_	
45	
- 6	
_ 9	
DRAFER LABORATORY JOVIAL STRUCTURE IN DIAGRAM OF TRANSFER-WETTEL	
4	
3.	
-	
5.5	

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF TRANSFER\*HRITE\*OUT

TRANS	Tons and the second sec
EXCEPT TRANSFER MRITE OPERATES ON PUTOUT	S TRANSFER'WRITE3. RANSFER'WRITE OUT ON PUTOUT
TE DE BUGG S	**************************************
MAX'FOUTPUT = MAX'M OUT-BLK) &	* MAX(MAX*FOUTPUT.**
OIFEITH CONTRACTOR	T 8+ T 8+ PUTOUT 1 PUTOUT 2 TO
	*OPEN INPUT PUTOUT*2 % * *OPEN OUTPUT PUTOUT*1 % *
	FOR I B 0. 1. MAX-FOUTPUT & ***********************************
	+-*ORIF 1 \$*POS(PUTOUT-2) = I +- +-*ORIF 1 \$*POS(PUTOUT-2) = I +- +-*ORIF 2 \$*POS(PUTOUT-2) = I +- +-*ORIF 3 \$*POS(PUTOUT-2) = I +- +-*ORIF 4 \$*POS(PUTOUT-2) = I +- +-*ORIF 5 \$*POS(PUTOUT-2) = I +- +-*POS(PUTOUT-2) = I +- +-*POS(PUTOU
E G GREEN THESE STREET	*SHUT INPUT PUTOUT*2 \$ *

++\*\*ORIF 1 & \*---\*POS (PUTOUT:1) = I ++

\*\*OUT-BUF-SIZE \$ \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* FOR I M 0, I, MAX'FOUTPUT 8 %---\*IFEITH %-+--I EQ DUT'BLK S %--\* 161\*\* 241 \*\*\*\*\*\*\*\*\*\* PUTOUT'2 .. C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAN OF TRANSFER"WRITE"OUT SHUT OUTPUT PUTOUT'S \*OPEN INPUT PUTOUT\*2 & \*OPEN OUTPUT PUTOUT\*2 & \* SAUT INPUT PUTOUT-1 8 + \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*OUT(MESSAGE, MESS\*SN) s \* SAGE = CAT(MESSAGE, T'MESS) \* \*NEW\*OUT = NOT NEW\*OUT & \*
\*EXTRA\*BLOCK = FALSE & \* .......... TF DE BUGG \$ . \*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\* RETURN S

			* (SF8) BNT T. TOOLO LOCATION TO THE (SF8)	**************************************
	100		100	-
	2		1907	3
	1 No.		15	LINE
	# # # # # # # # # # # # # # # # # # #		9	3
ESIGN DIAGRAM OF TRANSFER WRITE OUT	95 9 9 9 435 4 63			
ESIGN DIAGRAN OF TRANSFER WRITE OUT	4 0 0	To AZBONE	FOR J = 0. 1. OUT'BUF'SIZE - 1 *	
SFER	10.	¥.	- 3Z1S.	
7 2			- 80	
0	1:1		9	
AGRA	159 FROM 157 *		:	
8			•	

TO THE STATE OF TH

C S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRAPHER DESIGN DIAGRAM OF TRANSFER WRITE OUT

	**************************************
* 160 FROM 157 *	FOR J = 0, 1, OUT BUF SIZE - 1

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of transfer-write-out

T. C.

	OUTPUT PUTOUT'Z DUT'LINE(\$J\$)+OUTPUT PUTOUT'Z DUT'LINE(\$J\$)+
DESIGN DIAGRAM OF TRANSFER MRITE OUT  ***********************************	FOR J = 0, 1, OUT BUF SIZE - 1

~	
141	
=	
-	
-	
-	
~	
G	
•	
0	
2	
-	
=	
~	
W	
0	
-	
_	
•	
_	
~	
= .	
<b>3</b> F	
00	
3.	
w w	
-	
Œ	
- 4	
<b>&gt;</b> w	
04	
5 B	
> =	
2 3	
Z Z	
TRA	
TRA	
ATORY F TRA	
RATORY OF TRA	
ORATORY OF TRA	
ORATORY OF TRAI	
BORATORY H OF TRAI	
ABORATORY AH OF TRAI	
LABORATORY RAH OF TRAI	
LABORATORY SRAH OF TRAI	
GRAH OF TRAI	
AGRAH OF TRAI	
IAGRAH OF TRAI	
DER LABORATORY Diagrah of trai	
DIAGRAH OF TRAI	
I DIAGRAH OF TRAI	
N DIAGRAH OF TRAI	
GN DIAGRAH OF TRAI	
DRAPER LABORATORY IGN DIAGRAH OF TRAI	
SIGN DIAGRAH OF TRAI	
S DRAPER LABORATORY SIGN DIAGRAM OF TRAI	
G S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF TRANSFER WRITE OUT	

1	**************************************
162	FOR J = 0. 1, OUT BUF SIZE - 1 F COURT PU

E S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of update-files

PPROC UPDATE-FILES 8

		" "-LOOK AHEAD TO SEE IF CUR-REC MUST BE CONTINUED TO "- INCLUDE A TYPE 1 COMMENT.	erectes the contract of the co
"UPDATES FILE 3 184 ADDING A NEW ME ORD, OR CONTINUING AN OLD ONE) IN ACCORDANCE WITH THE INFORMATION IN THE NEWLY CREATED SET OF FILE 4 REGS.  "I RECORD." "I RECORD." "I RECORD." "I RECORD." "I LECORD." "I LECORD." "I LECORD." "I LECORD." "I LECORD." "I LET SECORD NUMBER OF A NEW " "FILE 3 REC." "I LET YPE OF THE CURRENT " "FILE 3 REC." "I LET SECORD." "FILE 3 REC." "I LET SECORD." "FILE 3 REC." "I LET SECORD." "FILE 3 REC."	CHAIN WITH THE NEMLY  CRAIN WITH THE NEMLY  CREATED FILE & RECORDS  STHITTPE = BOX-HAP(SIMT-TOKEN*  S S	FILTH	# 10

C S DRAPER LABORATORY JOYIAL STRUCTURED DESI\*GET\*F1\*RES 8 \* DESIGN DIAGRAM OF UPDATE\*FILE3

**************************************	•	***************************************	* A CARRY CO . COOR * C. HANTS WATER	THE PARTY OF THE P	<ul><li>・ はなりにはなるでは、これにはなり、では、これにはない。</li></ul>	THE PARTY OF THE P	THE LAST CONTROL	NO SCOPE CREATE	OUMNY BOX RECORD	The state of the s	•		A STANDARD OF THE PROPERTY OF	FINISH UP CUR'REC		5655665565565656565656565656565656		BEING TERMINATED .	ALPHO NEW MARKET OF THE SAME AND THE SAME AN	•	*REC) S) EQ O AND CUR"REC NO 03 ** 176*	•	SIFEITH SCUR'REC EQ 0 \$ S+	•

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAIMER Design diagram of update file3

POPPPROC'STACK(= CUR'REC) & OPPPLAYOU'INFO &	121			**************************************	EC)8) ***********************************		
		* MATEVER BOX WE'RE IN **	POOLD AM MEAN		*IF PAGE STACK TOP GR 0 AND *SIMT UN IT (SACCESSACUR REC) \$1	PPUSH PROC'STACK(CUR'REC) 8 PPUSH LAYOUT INFO 8 PUSH REC = F3 AVAIL 8 F53 AVAIL = F3 AVAIL + 1 8 ***********************************	*** NULL OUT POINTERS **  *********************************
		ORIF STHITTPE EQ COMENTS 8	**************************************				

```
*IFEITH *---*STHT'TYPE GQ CONTROL*1.8 *---*NULL*SCOPE = TRUE 8 *
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             *---CLOSE*REC(CUR*REC) $ *
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               *****************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 221
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       ++*ORIF 1 8 **--*CREATE'H'PTR'REGICUR'REC **
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      **********************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        *******************
                                                                                                                                                                                                                                                                                                                                                                         ********
                                                                                                                                                                                                                                                                                                                                                                                          *---*HESSAGE = 14HIUP SCOPE STARTI*
                                                                                                                                                                                                                                                                                                                                                                                                                           *OUT(MESSAGE, MESS*SW) $
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             **********************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     *******************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        *IF STMF UNIT (SACCESS3 (CUR'REC) *
                                                                                                                                                                                                                                                                               " NEW STAT UNIT STARTS SCOPE
JF A CONTROL PHRASE "
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         ***********************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   ************************
                                                                                                                                                                                              ********************
                    +H*PTR($ACCESS3(CUR*REC)$) = 0 *
                                                     *V*PT2(SACCESS3(CUR*REC)S) = 0 *
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    ********************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     *CUR*REC = NEW*REC s *
*INITIATE*RECORD(CUR*REC) s *
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      **********
                                                                                    *INITIATE "RECORDICUR" REC) $
*NULL" SCOPE # TRUE $
*WRITE3 # TRUE $
                                                                                                                                                                                                                                                                                                                                                                                                            •
                                                                                                                                                                                                                                                                                                                                                                         . IF DEBUGIO $ .
                                                                                                                                                                                                                                                                                                                                                                                                                             ************
                                                                                                                                                                                                                                                                                                                                                      ************
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 .......
C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DI* 8 MMER
Design diagram of update files
                                                                                                                                                                                                                                              --CRIFI/ STMT*UNIT($ACCESS3(CUR* * * *REC)$) /) GQ CONTROL*1 $
                                                                                                                                                                                                                                                                                                    **********************
                                                                                                                                                                                                                               **********************
```

C S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRAMER DESIGN DIAGRAM OF UPDATE FILES

|--|

\*ETURN S

G S ORAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of update-file3

. 166 FROM 167 \*

\*CREATE\*FILE, RECS & \* CONTINUE\*\* BOX (CUR\*REC) & \*

BEALTH DIVERTE OR TABLES FOR

C S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of update-file3

\* 169 FAOR 167 \*

THEETTH ********************************	* NEW REG) & * * * * * * * * * * * * * * * * * *		* ************************************	CREATE V PTR RECIOUR RECEPT	*LINES(BACCESS3(NEW*REC)\$) = 1 **  *LINES(BACCESS3(NEW*REC)\$) = 1 **  *SIMT UNIT(BACCESS3(NEW*REC)\$) **  *SIMT UNIT(BACCE
--	--	--	--	-----------------------------	--

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design Diagram of Update-Files

occessossesses	
•	
***************************************	
IL BACK VISACCESSSICUR RECUSO	
	***
•	BACK THROUGH A STUMP
*************************	
•	***************
	* POP LAYOUT INFO 8 .
*************************	
*CUR-REC = 1/ BACK *VISACCESS3 (CU*	
*R*REC181 /1 8	
***********************	
·	
*************	
*IF DE BUGIO S . sessessesses	
	TOWESS - IFORMATICUR RECT & .
	HIGACK THRU ) 8 .
*********** **************************	*HESSAGE = CATIMESSAGE, T.MESS) .
••	
*OUT (MESSAGE, MESS'SW)	MESS'SWI 8 .
*********	

A TO ESCH TO S

BESKEN GEFFER OF SERVICE SERVICE SAFRESTARY OF SLOW SIFEREY

LS S S S S S S S S S S S S S S S S S S	G S URFIER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF UPDATE-FILES	* 171 FROM 167 *		*IF BACK*A(\$ACCESS3(CUR*REC)*) *	!	**************************************		**************************************	***************************************		
--	---	------------------	--	-----------------------------------	---	--	--	--	---	--	--

CREATE'V'PTR'RECICUR'REC == NEW'REC) \$	
	# 7 2 # 10

のおおお のになるのでは、一般のでは、一般のでは、これをはのできるのは、これのはないのでは、これのはないのでは、一般のはないのでは、一般のでは、一般のでは、一般のでは、一般のでは、一般のでは、一般のでは、

G S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER DESIGN DIAGRAM OF OUT

PROC OUT(AA. GC) 8 .	**PUTS CONTENTS OF CHARACTER VARIABLE AA OUT TO THE ERROR FILE, IF CC IS 1, THIS FILE IS THE TERMINAL, IF NOT, IT	ITEM CE I 36 S S S S S S S S S S S S S S S S S S	officer 15)(AA) NQ 1H(I) S	PLACE ELACTOR OF THE PROPERTY	**************************************	*IFEITH *-4-**LN4 GR 72 \$ *TRMOUT(88, 72) \$ **  **CONTRACT **  **  **CONTRACT **  **CONTRACT **  **CONTRACT **  **CONTRACT **  **  **CONTRACT **  **CONTRACT **  **CONTRACT **  **CONTRACT **  **  **CONTRACT **  **CONTRACT **  **CONTRACT **  **CONTRACT **  **  **CONTRACT **  **CONTRACT **  **CONTRACT **  **CONTRACT **  **  **CONTRACT **  **  **CONTRACT **  **  **  **CONTRACT **  **  **  **  **  **  **  **  **  **	+=*ORIF LN¢ EQ 0 \$TRMOUT(13H(), 13)*  • **********************************
100 000 1110	VARTABL VARTABL FILE IS THE	TIER BE H	IF BYTE(SD	LIN = LN			

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMIER Design diagram of dut

FOR NEXT LINE:  - LN(\$) (554) =  - LN(\$) (554) =  - LN(\$) (554) =  - LN(\$) (554) =  - TORIF		*OPEN OUTPUT ZZZZZZ \$	**************************************	++++++++++++++++++++++++++++++++++++++	
---	--	------------------------	--	--	--

C S DRAFER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of Substr

*PROC SUBSTRIAL FIRST, NUM) 8 **	THE STATE OF
•	
THE SUBSTR	
. AA, INDEX 'FIRST', AND LENGTM	1
***************************************	
*ITEH FIRST I 36 S 8 *	Separate to the separate to th
PITER NUM I 36 S S	· 中華 中国 日本
*ITEM SUBSTR H 150 8 *	
*SF1 = AA \$	から 中 一 日本
**************************************	
•	
	· 在《中国· · · · · · · · · · · · · · · · · · ·
"IF BYTE(SO, 18) (SF1) NQ 1H(E)	THE VENTERAL
• •	**************************************
	A THE STATE OF THE
NUM LQ	
*******************************	*SUBSTR = SF1 S =
•	*********
	2 (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)
- 1 6K 1LM1 -	!
***************************************	+ S + OUTIC, RPIERR) S + OUTIC, RPIERR) S + OUTIC, RPIERR S + OUTIC, S + OUTI
••	*OUT(SFI, RPIERR) \$ *1C = 10H(LENGTH OF STRINGS ) \$*
(の)を) (の) (の) (の) (の) (の) (の) (の) (の) (の) (の	
- Constitution of the Cons	*ENCODE(6H(( IG)), LNI = TC6) *
	*1C1 = 1C6 \$ ***********************************
	*OUT(CAT(TC, TC1), RPTERR) \$ *
•	*ENCODE(GH(( IG)), FIRST * *
PARTIES DIFFERENCE ON CITY	
•	*TC = BH(LENGTH: ) \$ *

C S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRAMER Design diagram of Substr

******	
**ENCODE(6H(( I6)), NUM = IC6) **  ** ** ** ** ** ** ** ** ** ** ** **	
•	
0	6 A 0 B
_ ~ .	6.06 -
£ " .	
2 4 1	
~ ~ .	
100	
200 CATS	The state of the s
*ENCODE(6H(( I6)), NUM = TG. * 1C1 = TG6 * 0UT(CATIC, TG1), RPTERN * -0UT(CATIC, TG1), RPTERN * *SUBSTR = SF1 \$ **EUNSTR = SF	
I OFZ .	- 年
5 55 1 4	· · · · · · · · · · · · · · · · · · ·
W & WZ	
0 11011-01	
0 - 0 -	
Zeczza	
A POJUK+	
	2 7 2 2 1 1 2
	* 10 *
	12 1
	E
	*
	1 04
	1 - 1
	+ E5 N.
	+ 2Z+L+
	TE(\$6, NUM\$)(\$F2) = BYTE(\$5* FIRST, NUM\$)(\$F1) \$ * *  FIRST, NUM\$)(\$F1) \$ * *  BSTR = SF2 \$ *

ALE BULESCOP THE CREEK OF FALLS .

E RESULT OF  6 STRING AA  STRING AB  STRING		
STRING AA**  STRING AA**  (SF1) NQ 1H(C) ************************************	**************************************	
STRING AA**  STRING AA**  (SF1) NQ 1H(C) ***  CONST2) NQ 1H(C) ***  ***  CONST2) GR ***  ***  ***  ***  ***  ***  ***  *	•	
(SF1) NQ 1H(I)		
(SF1) NQ 1H(C)		
(SF1) MQ 1H(I)		
(SF1) NQ 1H(C) - SF1 = GNVERT(AA) S - SF2 = CNVERT(BB) B - SF2 = CNVERT(		
(SF1) MQ 1H(T) - SF1 = CNVERT(AA) &  (SF2) MQ 1H(T) - SF2 = CNVERT(BB) &  CONST2) GR - CAT = SF1 &  CONST2) GR - CAT = SF1 &  CONST2) GR - CAT = SF1 &  CONST2 CONCAT ERD &  CONST3 LN2  BYTE(\$6, LN		
(SF1) NQ 1H(C) -SF1 = CNVERT(AA) 8 - (SF2) NQ 1H(C) -SF2 = CNVERT(BB) 8 - (CONST2) GR - (CAT = SF1 & (CONST2) GR - (CAT = SF1 & (CONST2) GR - (CAT = SF1 & (CONST2) GR - (	1	
(SF1) NQ 1H(C)	88	
(SF1) NQ 1H(f)	* * * *	
(SF1) NQ 1H(f)	s 88 s	
(SF1) NQ IH(C)		
(SF1) NQ IM(C) SF1 = CNVERT(AA) &  (SF2) NQ IM(C) SF2 = CNVERT(BB) &  COMST2) GR CAT = SF1 &  COMST2) GR CAT = SF1 &  COMST2) GR CAT = SF2 = CNVERT(BB) &  COMST2) GR CAT = SF2 = CNVERT(BB) &  COMST2) GR CAT = SF1 &  COMST2) GR CAT = SF2 = CNVERT(BB) &  CONST3 CAT = CNVERT(BB		
(SF1) NQ 1H(T)		
(SF2) NQ 1H(C)	BYTE(\$0, 18) (SF1) NQ 1H([) .	:
(SF2) NQ 1M(C)	* S * S * S * S * S * S * S * S * S * S	• :
(SF2) NQ 1H(C)		
(SF2) NQ 1H(C)		
(SF2) NQ 1H(C)		
T) EQ D \$ "CAT = SFI \$ "  COUST2) GR " "CLAN E "  COUST2) GR " "COUST2 ERDING FR.  TC = LOH(TRUNCATE STRING FOUNTS FOUNTS FRER) \$ "OUTSF2, RPIERR) \$ "OUTSF2, RPIE	BYTE(\$0, 18) (SF2) NQ 1H(C) .	:
T) EQ 0 \$CAT = SF1 \$		• :
CONST. LNZ - SOURCES, RETERR) & CONST. LNZ - LNI + CONST. LNZ - COUNT. C. RPTERR) & CONST. LNZ - CONST. LNZ		
T) EQ 0 S **********************************	• •	
CONST. LNZ - CONST		
CONST2) GR		
CONST. LNZ - FOURTERS) & CONST. LNZ - CONST. LNZ - C		
CONST2) GR	のでは、 では、 では、 では、 では、 では、 では、 では、 では、 では、	
CONST2) GR	***************************************	
TC = 20H(*** CONCAT = RANGE   CONCAT = RANGE   CONCAT = RANGE   CONCAT = RANGE   CONCAT = LOUT(TC, RPTERE)   CONCAT = RANGE	+ LN2 - CONST2) GR .	
**************************************	THE STATE OF THE CONCE	CONSIZ &
*OUT(TC, RPIERR) \$ *TC = L@H(TRUNCATED STRING!) *OUT(TC, RPIERR) \$ *OUT(SF2, RPIERR) \$ **CONST, LN2 - ** **SYTE(\$6, LN2 - ** *	••	
FOUT(TC, RPTERR) S FOUT(SF2, RPTERR) S CONST. LN2 - F BYTE(\$6, LN2 - F		
CONST. LNZ -	E 142 × 8188125×	•
CONST. LN2 - F	*OUT(TC, RPTERR) \$	
CONST. LN2 -	**************************************	
CONST. LN2 - 8 BYTE(\$6, LN2 - 8		The second second
SHEET PRINCE THE TRANS	CONST. LNZ - BYTE(\$6, LNZ -	
	TORREST AND ADDRESS - THORN	

BCRES BASESSEN ON DRACKS.

C. J. GRUNDE FFEMENTALOSK TORING ELINGISED BESTER DEPOSTARE

PROC CNVERTIAN 8 .

***************************************		
ITEM AA H 150 8 .		
ITEN CNVERT H 150 8 *		
m		
ITEM II I 36 S 8 .		
9		
*****************		
CNVERT CONVERTS NORMAL		
IAL CHARACTER STRINGS		
STH 150 TP A DESCRIPTO		
RH SUITABLE FOR EXTEND		
OPERATIONS. CNVERT		
IMPLY RETURN		
GUNENT IF THE STR		
READY CONVERTED. STR		
ONTAINING MORE THA		
ARACTERS WILL BE TR	8	

PRETURN S  PRETURN S  PARTOR S  PART		**************
· · · · · · · · · · · · · · · · · · ·	S. CAVE	ERT = SF3 &
		JKN S
AA3(808) = 6H( ) 8 ** **PREVENTS OVERFLOW OF BVTE ** II = 1 \$ 50 = 1H( ) 8 **CLEARS OUT CHARACTER STRING**		
AA1808) = 6H( ) 8 * PREVENTS OVERFLOW OF BYTE  II = 1 \$  NONE = 0 \$  FG = 1H( ) \$ **CLEARS OUT CHARACTER STRING***		
LATER  LATER  II = 1 \$  NONE = 0 \$  SFG = 1H() \$  CLEARS OUT CHARACTER STRING*	*************************	
LATER ***  [1 = 1 \$ * * * * * * * * * * * * * * * * * *	* PREVENTS OVERFLOW OF BYTE *	
II = 1 \$  ONE = 0 \$  SF6 = 1H( ) \$  CLEARS OUT CHARACTER STRING*	LATER	
ONE = 0 \$ ** F6 = 1H() \$ ** CLEARS OUT CHARACTER STRING*	•	
SF6 = 1H() 8CLEARS OUT CHARACTER STRING.	)ONE = 0 \$	
**CLEARS OUT CHARACTER STRING*		
•	** CLEARS OUT CHARACTER STRING*	

S A A S

C. S. DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRAMMER DESIGN DIAGRAM OF CNVERT

	**************************************	** FIRST NONBLANK CHARACTER HAS  ** BEEN LOCATED EXACTLY **	**************************************	PICK UP THESE FIRST FEM . WORD-SKEWED CHARACTERS**	*IF II NO 24 \$  **AVID GIVING BYTE FUNCTION A ** 162*  * ZERO ARGUMENT************************************	PICK UP REST OF STRING	S CT - 31 - 32 - 41 - 13 - 4 - 4 - 4 - 4 - 4 - 4 - 4 - 4 - 4 -	ALF LNG GR MAKCOL + CONST & LOUG	**************************************
IF SAMURIAN NO 61 CONTROL OF SAMURIAN NO		••••	•••••		•••••	••••		* * * * * * * * * * * * * * * * * * *	Andreas as as on the properties
DESIGN DIAGRAM OF CNVERT  ***DO WHILE COONE EQ 8 AND II LQ ***  ***241			•••••	••••				••••	S SEEM O CHOSEN ON CHARLS ON BUT 228 DICENS

1000	
~	
-	
DIAGRAME	
-	
Contract of the last	
~	
1000	
•	
-	
-	
-	
_	
0	
2000	
0	
-	
-	
Participant of	
-	
Company of the contract of the	
and the	
•	
L STRUCTURED DESIGN	
=	
N S	
IALS	
IAL S	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
3	
C S DRAPER LABORATORY JOYIAL S DESIGN DIAGRAN OF CNVERT	

TI = II + I &		
IF II EQ 25 AND DONE EQ 8 *****SF6 # SPACES(HAKCOL) \$ .	143	
CHVERT = SF4 & 0	*	

A CONTRACTOR OF THE CONTRACTOR

1985 TAR

		:		•	:
		********************	_	TRUNCATE THE STRING	*********************
		:	*LN6 = MAXCOL + CONST &	9	:
		:	S	8	:
		:	8	ST	:
. 163 FROM 180		•		w	:
		•	2	=	•
		:	Š	H	:
: 2:		:	1	3	:
: ":		:		3	:
2:		:		Z	:
: ":			3	:	:

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of spaces

TPROC SPACES (NUM) & T		**RETURNS AN EXTENDED CMARACTER STRINS CONTAINING	THE NUMBER OF SPACES DEFINED BY NUM".	OITER NUM I 36 S & CONTROL SPACES M 150 S CONTROL SPACES M 150 S CONTROL SPACES		*OUT(TG. RPTERR) s
PROC SP	••	CHARA	THE NUMB	ITEM NUM	 TF NUM G	•

C S DRAFER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER Design diagram of mull

			erreserves		
**************************************	* **RETURNS TRUE IF STRING AA IS * A NULL STRING**	STER AA N 150 B • STER NULL B S • SF1 = AA S	IF BYTE(SQ, 12) (SF1) NQ 1H(C) .	**************************************	OCCUPANT CONTRACTOR OF CONTRAC

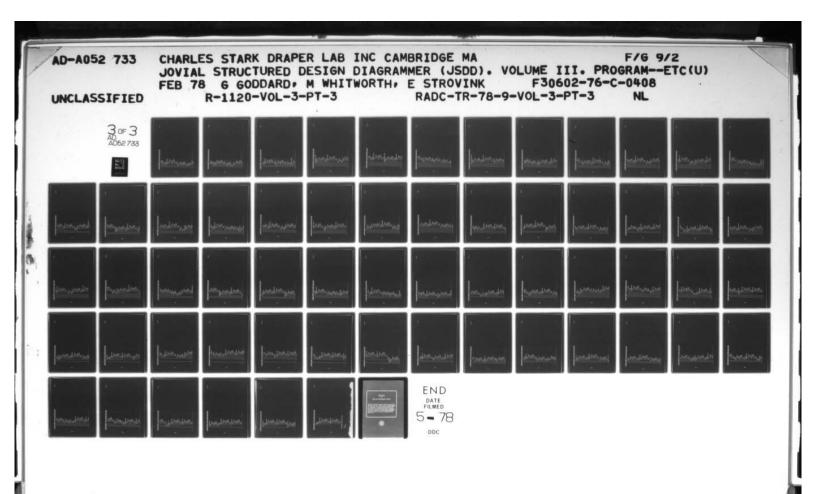
G S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER Design diagram of length

			**************************************
PROC LENGTHCAN S	**RETURNS THE LENGTH OF STRING AA**	TIEN LENGTH I 36 S & COLLECT A M 150 S COLLECT A	

PLENGTH INT - CONST & .

25 V CC 2 001 L 3 01 M M V 3

ITELY SELF-REGURSIVE		PAIN	-BYTE'EN	RENGUO+	ţa;	CHVERT	· · SPACES	100	CHVERT	TRHOUT.	 CREATE FILEY TECS	ACCESS2	car	CHVERT*	001	IF ORMAT	CNVERT	ENCODE+	•001-	RENGUO+
YER!	1063	1	ŀ.				• •	• •		• •			• •	•	• • •		•	• •	• •	



***************************************	GNVERT		E MQUO+	· · · · · · · · · · · · · · · · · · ·	CUVERT			•3000E•	XWI.	-100	647	CNVERT	• 100	IFORNAT	· · · CH VERT*	ENCODE+		out-	· ·SPACES•
---	--------	--	---------	---------------------------------------	--------	--	--	---------	------	------	-----	--------	-------	---------	----------------	---------	--	------	------------

E S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRANNER Invocation Diagran of the Design Diagram Generator (DDG)

CHVERT*			*100	 7.484		CAT		18 0 May 18 18 18 18 18 18 18 18 18 18 18 18 18	• • • • • • • • • • • • • • • • • • • •	FORHAT	· · · · · · · · · · · · · · · · · · ·	· · · CHVE 2T ·				4 7 1 2 1 2 1	- udboo	**************************************	Tenenal Control of the Control of th	2000年の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の	*CNVERT*		A CONTRACTOR OF THE PERSON NAMED IN COLUMN TWO IS NOT THE PERSON NAMED IN COLUMN TWO IS NAMED IN COLUMN T	-INITIALIZE	-0.1	A STORY OF THE STO	- A NOT THE STATE OF THE STATE	ACCESS3	A CALMINON A		· · · CHVERT	X A STATE OF THE S	6.3600000000000000000000000000000000000	IFORMAT	中級問題是發展了	CNVERT	• • • • • • • • • • • • • • • • • • •	AND CONTROLS TANKED ENGINEED	**************************************
•	•	•	•	• •	•	•	• •	•	• •	•	•	•	• •	•	•	• •	•	•	• •	•	•	• •	•			•		•		•	•	• •	٠	•		•	THACORES	S S SSS SS	•

	· · · · · · · · · · · · · · · · · · ·										*193			5		AU 10 10 10 10 10 10 10 10 10 10 10 10 10				d				ERS		五百二五年 五年 五百二
		R- WRITES	121.25.00	CHVERT			CHVERT*	ENCODE -			36	1100001	21.				•30	4.7		183470	A 12 10 10 10 10 10 10 10 10 10 10 10 10 10			PAGE NUMBE	22,000	在 * 2 位的名称形式
-00T-	-REMQUO+	TRANSFER-		3	OUT	IFORMAT	•	9			DVANCE. PAS	CAT	CHVER		-IFORMAT	CHVERT	ENCODE .	-0UT+		* * *	-CNVERT*	OUT	CNVERT	COMPUTE. PA	-CAT	· 等質 的配品等 在在 · · · · · · · · · · · · · · · · · ·
		• •	* *		F 7/8	* *		4 4			54-	• •	•••	• •		• •	•		. !	- 64			3-	9		•
e 6	•	•	• • •	•	•	• • •	•••	•	•	•	• •	• •	• •	••	•	•	• •	• •	••	• •	• •	• •		• •	••	•

G S DRAPER LABORATORY JOYLAL STRUCTURED DESIGN DIAGRANNER INVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

CHVERT*	ENCODE+	*001	RENGUO.	TRANSFER- WRITE- OUT	car	CNVERT*	•10	IFORMAT	· ·CHVERT*	**************************************	** California 4		ENERATE CONTENTS - HEADER	Access out	car	· · · · · · · · · · · · · · · · · · ·	•0ur•	IFORMAT	· ·CNVERT*	ENCODE+		REHQUO+	* TRANSFER - METE - OUT	car	· · · · CHVERT	*100	
													3		:	::	• •	• •	•	•	•			•		• •	
					194.19		10.10	4 8												1.5					1.4		
• •																Beer Co.											
•		•	•••	•••	• •	•••	•		1 5				10					S LOW		24.5			10		a 8		
				::	::	::						• •	• • •														
9-1							6 - p					••	•		••	•	• •	• •	•	•							•

G 8 DRAPER LABORATORY JOYTAL STRUCTURED DESIGN DIAGRANNER Envocation Diagram of the Design Diagram Generator (DDG)

	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·		· · · · · · · · · · · · · · · · · · ·	uxx	· · · · · · · · · · · · · · · · · · ·	·0uT•	を 一	ADVANCE PAGE						· ·TFORMAT	The state of the s	· · · · · · · · · · · · · · · · · · ·	1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	· · · · · · · · · · · · · · · · · · ·	2000年7·	out	ECTITS, DO.	REMQUO+	- CENTRELE CONTRACTOR	CNVERT*	0720000	IFORMAT	NAME OF THE PERSON OF THE PERS	- CANER	• •ENCODE+	の の の の の の の の の の の の の の の の の の の	LENGTM	\$4885772*	· ·CNVERT				CNVERT*	が記録と 調報の 東京 不の以前の数数本数明をあって	ENCODE+	1000の日本本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本	-LENGTH		* 1000000000 · · · · · · · · · · · · · ·		の 一	001-	AND THE REAL PROPERTY OF THE REAL PROPERTY OF THE PARTY O	REMOUO+
							•							•					•						•													•		•	•				:	•	•		•
- 18							-	1	*		*	3																																					
	10	*	Ğ	9		9	3		180						ü	Į.		Ū			Ī	ā				*						la.		8									6					1	i
	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	• '		ı	•	•	•	•	•	• '	ı		•	•	•	•	•					•		
																			•	•			•										•	•					•	•	•		•		•	•	•	•	•
																																																Same Person	

G S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANNER ENVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

GENERATE CONTENTS - HEADER	ACCESS OUT			· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	TFORMAT	The second secon	· · · · · · · · · · · · · · · · · · ·	ENCODE+	1 - 1 - CAMENTA	. 1.00	REMQUO+		TRANSFER- WRITE - DUT	car	\$ ADmx 8			CALCULATION 15.2	*8 GWE 5 2 m	· · · · · · · · · · · · · · · · · · ·	ENCODE+	The second control of	 	ADVANCE PAGE		 CNVERT*	4 May 2 30 Cla	1000000000000000000000000000000000000	IFORHAT	CNVERT*	P. S. Seller	S. C. C. C. C. S. S. S.	•100	REMQUO+
		::	•	•	•	• •		•	•	•	•	•	•	•	•	•	•	•		•	• •	•	•			×	:	•		: :	. ,	•	•	::	2.5
* *											•									2 8						b							2 .0		0
	COLUMN TO SERVICE STATE OF THE PARTY OF THE			-							-			-								24.0	-			-	-	-	200			-0.0	100	-	

											# 2 1 1						document.		100000000000000000000000000000000000000		2. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.				1000	1000				MR ITES			ER TO	
がはなりして 大	CHVERT	*IFORMAT	#3-Ex	CHVERT	- Sections	UD NA. C. L. C.	LENGTH	· ·CHVERT	一川大小田をオーラー		-IF ORMAT	CHVEDTO	Barren .	ENCODE+	OUT*	SPACES	COMMECT BOKES	-ACCESS3	-641		CAVERI	• 1F0	IFORNAT	· ·CNVERT	· · · · · · · · · · · · · · · · · · ·			0u1•	REMQUO+	TRANSFER WRITES	CAT	10 - 2 - C - C - C - C - C - C - C - C - C	CNVERT	
		•	•	• •	•	•	•	•	•	• •		• •	•	• •	n n		-		•	• •	•	• •	•		• •	• •	•	• •		•	•	• '	2 4	
				4		•	•				W 70			No.		•				, ,,								•						

G S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DEAGRANNER INVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

 - MAX SPEEs	-100	CCESS out	CAT - v. Ecus, aut	CNVERT	 IFORMAT	CHVERT	· · · · ENCODE ·	0uT-	REHQUO+	TRANSFER WRITE OUT	CALL CALCALLY	CHVERT	out-	IFORMAT	CNVERTO	ENCODE+	nax	• 100	 CNVERT*	OUT* COA	DRAW'LINE	· ·ACCESS OUT	
		• •	::	::	•	•	•						•		•		::	::					

8 S DRAFER LABORATORY JOYTAL STRUCTURED DESIGN DIAGRAMER INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

		· ·our		IFORNAT	Cuve.70		ENCODE+		RENGUO+	*** TAANSFER WRETE OUT		CAF	CHVERT	A STATE OF THE STA	out		IPORTAL	· · · - CNVERT*		ENCODE+	ux		•0uT•		用級部 性形 中	CNVERT			DASHES		*CAT		CNVERT	 N. S.	IFORMAT	· · · · · · · · · · · · · · · · · · ·	A STATE OF THE PARTY OF THE PAR	ENCODE+	4 25- DEPLOY GIRCHN CONTEN	A TONING TONING TO BE TONE
•		•	•	• •		•			• •		•	• •		•		• •			•			•			•	•			•		•	•	•					•	• •	
																									-	4 0				1	13									8
	10								0.00				7							90	1	0	+ 2				-	*	4 0	1			-	10		1	4.13			1
•		•	•	• •		•	• •	• •	• •		•	• •	•	•	• •	• •		•	•	• •		•	•	• •	•	•	• •				•	•	• •		• •			•	٠.	0
													4																* *				2			100			10 10 10 10 10 10 10 10 10 10 10 10 10 1	1961年 1453

C S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRANHER INVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

REMQUO+	TRANSFER. WRITE. OUT	-CAT	CNVERT*	Sept.		IFORMAT	2000年間の	CNVERT*	1 10 日本日の日本	• • • • • • • • • • • • • • • • • • •	 1 and a second	•00T•	IFORMAT	* Contract of the Contract of	· · · · · · · · · · · · · · · · · · ·	· · · · ENCODE ·	4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	第一条图100mm	OUTPUT HEADER	AGGESS OUT	179		CNVERT*	**************************************	78 (SC)	IFORHAT			ENCODE+	 1. (1. (1. (1. (1. (1. (1. (1. (1. (1. (	REMQUO.	- 国際財政を記する。 - 本 ・ 本	TRANSFER WRITE OUT	7 CAT	1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1		A DE THE SEZP DUT ENT ED HER THE	CONTRACTOR OF SELECTION CONTRACTOR CONTRACTOR
													Un To																									0
																																					1	
				•	•	•	•	•	•	• " •	•		•	• •		<b>»</b> •		•	•,•	•	•	•	•		•		•	•		•	•0	•	•	•	•	• •	東京 日本記書	

6 8 DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRANNER INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

**************************************	 ·		THE STATE OF	ENCOUE+	一年 大学 のいました 大学 のいました 大学		A THE PROPERTY OF		***CMVFP T*	Appendix of the Apple	ouf*	- 10 Contract of 10 C	-CNVERT*	大田 はいないかにはをする 日日	-IFORMAT	CNVERTO	•== 5.0000	- CHUCOLE	-LENGTH	•	CNVERT	-001-	SPACES"		REMQUO+		· · · · · · · · · · · · · · · · · · ·	-CNVERT.	* * * * * * * * * * * * * * * * * * *	ENCODE +	de de la companya de	A set The Shady I	CNVERT*	おから かりの		中国 医多种性 医多种性 医二甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基甲基	A 9			RANSFER-WRITE3	A でき 信仰 は 200mm を	-c41	CNVER TO THE STATE OF THE S	TARREST SERVICE SECTION OF SECTION
																 			i						E.	IFORMAT		Ş		Ž	LENGTH		Š			003470	2	007		3		١.		
																 									•	JE.					-LE		i.		-001	1	Ē.			į				
																 																			١.									
		•					•					•																			•	•	•					•	•					
	•						•	•	•			•	•			 		•						• •		•	•		•			•					•	•		•			•	

G S DRAPER LABORATORY JOYTAL STRUCTURED DESIGN DIAGRANHER INVOCATION DIAGRAN OF THE DESIGN DIAGRAM GENERATOR (DDG)

-	OUT		IFORNAT	CNVERT	ENCODE+	HAX		TRANSFER WRITE		CAT		· · · · CNVERT*	•	out.	IFORNAT	· · · · · · · · · · · · · · · · · · ·	·	ENCODE+	HAX		001-	PENDINO +		2	ACCESS 001	caT		CNVERT	-100	IFORHAT		· · CNVERT*		ENCODE+			RENQUO+		TRANSFER WRITE OUT	A "S-CAT ABOUT 2525 S. DEPT	
			• •		 			!						-									0	STARS		Section 5.	•				•					•			Spirit de		
					 															•					 				 												
	•	•	• •	 			 	• •	•	•	•	•				•			•	•	•		•	• •	 	•	•	• •		•	•		• •		•		•	• •	N. 1. 1. 10	• •	

G S DRAPER LABORATORY JOYTAL STRUCTURED DESIGN DIAGRANNER IMVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

THE RESERVE AND A SECOND SECON	· · · · · CHVERT*		-Trowns	· · · · CNVERT•		Encode	HAX		•0ur	CONSTRUCT - LINE					-CNVERT*			-IFORNAT		CNVERT*	ENCODE+		100000		-KENUO3+	-TRANSFER-WRITE.OUT		-CAT		CHVER I	DUIT		IFORMAT		CNVERT*		EMCOUE+			-0uT-		THE TAX BEST OF THE SAME STATES	ESS. OUT THE THE PART OF THE P	
										TRUCI		-ACCESS-001	CAT	•				116	•	١.	:		001		- KE	-18	•		• •		•	•		•	• •	. •		:	•		ACME		ACC	
										5		•	•	•	• •	•	•	•	•	• •	•	•	•		•	•															-		•	
										1	• •	•	•	•	• •	• •	•	•	•	• •	•	•	• •	• •	•	•	•	•	•	•	•	•	•	•	• •	• •	•	•	•	• •	•		•	•
	• •	• •	•		• •		•	•	• •	•	• •	•	•	•	• •	• •	•	•	•	• •	•	•	• •	• •		•	•	•	•		•	•	•	•	• •	•	•	•	•	• •	•	•	•	
		 					•	•							• •		•	•	•		•	•					•				•	•	•	•				•	•			•	•	•
																																										1		
																																										100		

G S DRAPER LABORATORY JOYLAL STRUCTURED DESIGN DIAGRANMER ENVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

CAT	·		IFORMAT	CNVERTO	ENCODE+	.100	REMQUO+	*** TRANSFER WRITE OUT	CAL	CNVERT*	011	FORMAT	CHVERT	**************************************		*	LENGTH	A WOODS IN THE STREET	CAVERI	HIM an accordance	-00T* * ********************************	OUTPUT HEADER	ACCESS OUT	173		· · · · · · · CAVER T*	· · ·00T*	· ·IFORMAT	THE RESERVE THE PROPERTY OF THE PARTY OF THE	
																	. !			Ī.	Ĭ.	Ĭ								
		٠.			•																									
	•													• •	• •										•			• •		
																													Marchine A south	

G S DRAPER LABORATORY JOYLAL STRUCTURED DESIGN DIAGRAHHER Invocation diagram of the design diagram generator (DDG) -- TRANSFER - WRITE - OUT --CNVERT ---CNVERT\* ---nax --IFORNAT ---OUT. . .--ENCODE+ ---CNVERT -------IFORMAT ----------- OISPLAY'STUMP'REF --- RENGUO. ---ENCODE + ---REMQUO+ ---CHVERT \*--SPACES\* -100------OUT----ACCESS DUT

おかがら 下田

. .--CNVERT\*

C S DRAPER LABORATORY JOYLAL STRUCTURED DESIGN DIAGRANNER Invocation Diagram of the Design Diagram Generator (DDG)

			Chief also				LITE- OUT		FK. HKIRCACOL					£	ğ.														•	The state of the s	•				
************	IFORNAT	CHVERT		+3000K3	OUT.	 	TRANSFER-W	CAT		CMVER	-TUO	-	T- IT- OKTA	CHVERT	ENCODE	A CONTRACTOR	×#	·100	CAT	100000000000000000000000000000000000000	CHYERT	•100	DA SHES	A	- uccess out	 CHVER	 	IFORMAT	CNVERT		ENCOOE	•TC0		· Nemana	
	•		* * * *		•	• • •	•		•		• •		•	•	•	• •						•		• •		•		• •				•	• •		C & UNIVERSAL ST

G 8 DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRANNER Invocation Diagram of the Design Diagram Generator (DDG)

TRANSFER WRITE OUT	CAT	*Telunger		• • • • • • • • • • • • • • • • • • • •		IFORNAT	· · · · · · · · · · · · · · · · · · ·	1000年の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の	ENCODE+	• 17			2 3 Kg ath	ORAN"LINE	6 COUNTY - COUNTY	ACCE 55 - 00 I	CAT		· · · · · · · · · · · · · · · · · · ·		F IF ORNAT		** ** Calcade An	+3000N3		A PAID OF A STATE OF A	KENEDO	TRANSFER WRITE OUT	* ** I KYP. C. S. NY TEE, ONL	· · · · · · · · · · · · · · · · · · ·	* - CV N CD - * -	•100	· · · · · · · · · · · · · · · · · · ·	A	· · · · · · · · · · · · · · · · · · ·	4 3 4 5 - 5 - 5 - 5 - 5 - 5 - 5 - 5 - 5 - 5	- Thomas ENCODE	A THE COLUMN TO SERVICE A SERVICE AS A SERVI	•001•	FM DE ÎNE DESTON DEFENSM GENERAL
	•	•	•	•	•	•	•	•	•	•	•			•	•		•	•	•	:	•	•	•	•	•	:	•	•		:	•	•	•	:						
	8.7	1	•				•	•		•		١		-						 VI.														-				1 18		0
			•					•								2 1					×									4								* *		1001

E S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMER INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

car	A P P P P P P P P P P P P P P P P P P P	* * * * * * * * * * * * * * * * * * *	·0uT•	DASHES	· ·ACCESS OUT	Table of the state	· · · · · · · · · · · · · · · · · · ·	CNVERTO	· · · · · · · · · · · · · · · · · · ·	** * * *** SANGERS	IFORMAT	· · · · · · · · · · · · · · · · · · ·	***************************************	ENCODE -	ouf-	9.500 DR3 " " - ENCODE 6		TRANSFER" WRITE" OUT	- CAT OSSES	William Co. St. A. St. A.	**************************************	011	IFORMAT	· · · · · · · · · · · · · · · · · · ·	CNVERT	+3000E+	**************************************	# 3000M2mm* * * * *		FORMAT	1800001-44 100	· ·-CHVERT*		a a a seedingston	00.	*OUTPUT*HEADER		
	• •	•	•	• •	•	• •	•	• •	•	• •	•	•	• •	•	•	• •	•	• •	•	• •	•	• •	•	• •	• •	• •	• •	•	٠.	•	•	• •	٠	•	• •	•	•	
										• •			• •		•	• •	•	• •	• •	• •	•		•	• •	• •	•		•	• •	•	•	• •		•				• •
•	•		1		1.10		- 9	8.1				- 36	7	3.400																					96.714			

G S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGNAMER INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

							•		•					WEITE-OUT			ERT.			* 115,00£	ERT.	-																					
CAT		CNVERT			TEOPHAT	34.0 mm	CHVERT	***************************************	ENCODE	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		PENDUD.	SAUCE COMP.	TRANSFER WE	CAT		CHVER	THE PERSON AND ADDRESS OF THE PERSON ADDRESS OF THE PERSON AND ADDRESS OF THE PERSON ADDRESS OF THE PERSON AND ADDRESS OF THE PERSON AND ADDRESS OF	 IF DRNAT	第一日本の日本の日本の日本	CHVERT	300000		HAX	* * SACSON *	 car	TANKSOND	CNVERT		AL STANSFORM	CHVERT	TEODMAT	- V - C - C - C - C - C - C - C - C - C	CNVERTO		- EMCOUE.	***********	Ward-Strate	CNVERT*			SPACES.	
																	•			•				•			•			•				•			•	•	•		•		•
	• •	•	•				•	•		• •		•	•			•	•			•	•		•	•	•	 •	•	• •		•	•	• •	•	•			•	•	•		•	•	•
•	•	•	•	• •		•	•	•		• •		•	•	• •		•	•	• •		•	•		•	•	• •	 •	•	• •		•	•	• •	•	•		•	•	•	•	• •	•	•	•
•	•	•	•	• •	•	•			•	•	• •	•	•	• •	•	•		• •		•	•	•	•	•	• •	•	•	•	•	•	•	• •	•	•	• •	•	•	•	•	• •	•	OCAS TOR	Charle brand

E S DRAFER LABORATORY JOYLAL STRUCTURED DESIGN DIAGRAHMER Invocation diagram of the design diagram generator (DDG)

RENGU3+		IFORMAT	· ·CNVERT*		ENCODE+		The state of the s	CNVERT.			PHZERR	***************************************		TRANSFER-WRITES		SAT S-CHARKE	· · · · · · · · · · · · · · · · · · ·			IFORMAT		CNVERT	ENCODE+		H4X	• • • • • • • • • • • • • • • • • • •	- IKAMSTEK BRILLE	CAT		CNVER TO		IFORMAT		CNVERT	 EMODILE .	HAX	•	.100	- OF HOUSE	Maria Caraca Car
			4 /	0	-	20	4.					0			i, v			ំ		100	•						•			10	•		•	• •	•	•	•	٠.		-
•	•	• •	•	•	• •	•	•		•	• •	•	•	• •	•	•	• •	•	•	• •	•	•	• •	•	•	• •	•	 •	•	• •	• •	•	•	•	• •	•	•	•			•

G 8 DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRANNER Invocation Diagram of the Design Diagram Generator (DDG)

STARS	ACCESS-OUT	 · · · · · · · · · · · · · · · · · · ·	 IF ORMAT	CNVERT	ENCODE	•Tr0	·REMQUO+	. TRANSFER WRITE OUT	- Cat	· · · · CHVERT	• 100		CMVERT*	ENCODE -	A SAN DERE	-T00	EXTRACT TEXT		cat		CAVERT	• Tuo	IFORMAT	• • • • • • • • • • • • • • • • • • •				
					4 0					A - N	•		• •				* *	* 1		•		•		•	٠	• •	S. Saydella	•

C S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRANNER Invocation Diagram of the Design Diagram Generator (DDG)

	······································	REMQUO+	-ACCESSA	A 2000 St. W	CAT	· · · · CAVERTO	· · · · · · · · · · · · · · · · · · ·	001.	**************************************	IFORMAT	CHVFPT*	<b>新品品的</b>	ENCODE+	HAX	4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	0UT*	REMQUO+	TRANSFER- WRITEL	11、12日、日本日本	car	· ·CNVERT	001-	STRAIL DEC COCTURALS	-30,000	CNVERT*	ENCODE+	nax	日本の日本	-100	-CAT	*CNVERT*	 - CHARBERTS	-IFORMAT	***CNVERT*	* THE SECOND	CITA S	-LENGTH

E S DRAFER LABORATORY JOYLAL STRUCTURED DESIGN DIAGRAMER ENVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (006)

- SUBSTR - CAVERT - CAVERT - CAVERT - CAVERT - CAVERT - CAT - CAT	
• • • • • • • • • • • • • • • • • • • •	
	 •
•••••••••••••••••	 •

																															CONTRACTOR O
+300€			April 100			8			•			•		•				WRITE3		PETRICLIPME.	ERT	•		58.70	4	•300					
ENCODE	HAX	-100	ENGTH	CNVERT*	<u>.</u>	OUTPUT-80X- 80TT	-ACCESS3		· ·CNVERT	• TUO	IFORNAT	CAVERT		3000m3	nax	ouf•	REMQUO+	TRANSFER - WRIT		1378	CHVER	· 00T	IF ORMA	CHVER TO	S WYS - SAME	DN3	******	THOUSE.	To the state of th	ACCESS-OUT	CAT
		*	7.			3.		• • •	• •	•	• •	• •	•		• •	• •	• •	• •	• •	•	• •	•	• •	• •	•	• •	•	• •		٠.	•
-										- Pick 1984	-0.0	-	0.12	-			2001														

.

6 S DRAPER LABORATORY JOYTAL STRUCTURED DESIGN DIAGRANNER INVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

	CAVERIT		 IFORMAT		· ·CAVERT	一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一	ENCODE+	日本 一 一	**************************************	2 NOBEL 1	REMQUO+	30.00	TRANSFER - HRITE - DUT		CAT	and Street Characters	· ·CNVERT*	State of the late	•100	· · · · · · · · · · · · · · · · · · ·	PIFORMAT	The state of the s	CNVERT	The second secon	ENCODE+	 (1) (1) (1) (1)	•1co•	4 A*******	-CAT	# # # # # # # # # # # # # # # # # # #		***************************************		-DRAW'LINE		ACCESS OUT		· · · · · · · · · · · · · ·		· · · · · · · · · · ·		· ·IFORMAT		· · · CNVERT*	· · · · · · · · · · · · · · · · · · ·	ENCODE+	O OUT. ACCT. SELECTIVE OFFICE	CARRY SEATER BLANCINGER DEALS
•					•	•		•		•	•	•	•	•	•	•		•	•	•	•		•				•	•	٠.			•	•	:			 •				•	•	•					
									•	•	•	•		•	•			•		•							•					•	•															
•	•	•	 •	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	• •		•			•	•	•	• •		•	•	•	•	•	•	 •	•	•	•	•			•		• •	MAR ES	CHKT 6-60

E S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER ENVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

REMQUO+	TRANSFER WRITE . OUT	CAT	- The second sec	 •001•	TFORMAT	The second second	CNVERT	THE COUNTY OF TH		HAX	* Section of the second	100-	CAT	4 2 4 4 4 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4	CNVERT	001-	A STATE OF STATE OF	-UASHES	ACCESS-OUT	CAT	30	 001*		CNVERT*	***************************************	- Encode	•0UT•	TRACE TO STATE OF THE STATE OF	KENGOO	TRANSFER. WRITE. OUT	被動物による・	 CNVFRT		•100	
																		١.																	
• •	•					•			•																		• •			•			•		
	• •																																		
						•																													

6 S BRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANMER INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

· ·CNVERT*	A	· ·ENCODE+						20 m 9 c 2 c 2 c 2 c 2 c 2 c 2 c 2 c 2 c 2 c		1.000	CHUEKI	2006	2000			OUTPUT - HEADER		ICCESS-OUT		-CAT	CNVERT			IFDRMAT		CNVERT	•	ENCODE+		001-	- 65 40110 -	KEN400*	TRANSFER WETTE OUT	•	CAT	75.8	· · · · · · · · · · · · · · · · · · ·	011	-1001-	IFORMAT	- Contractor	· · · · · · · · · · · · · · · · · · ·		ENCODE +	HAX		0011	TOTAL OFFICE PROPERTY OF	
									THE OWNER			BMCORE		OUT.	i	OUTPU		AC	•	•	 	•	•		•	•	•	•	1100									•		•				•	•		• •	10000	
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	• •		•	•	•	•	•	•	•	• •	•	•	•		•	• •		•	•	•	•	•	• •	•		•	•	• •	•	
•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	• •	•	•	•	•	•	•	•	•	• •			•	•	•	• •			•	•										
•	•	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•				•	•	•	•	•						•														. ,		
•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	• •	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	• •			•	•	•	•	• •		Del 7 1. 10.

G S DRAPER LABORATORY JOYTAL STRUCTURED DESIGN DIAGRANNER Invocation Diagram of the Design Diagram Generator (DDG)

 CNVERT	ENGTH	 KENDUO+	IF ORMAT	· · · · · · · · · · · · · · · · · · ·	ENCODE+	 001-	RENGUO+	**************************************	ACCESS OUT	CAT	· · CHVER T*	0uT-	IF ORHAT	· · · · CNVER T*	ENCODE+	001	REMQUO+	TRANSFER"HRITE" OUT
 		 				 												300

E BRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRANNER ENVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

|--|

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANNER ENVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

STARS	18.	· · · · · · · · · · · · · · · · · · ·	•00	IFORMAT	CNVERTO	ENCODE	our	 TAANSFER'WRITE'OUT	A P A STANDARDS	1	· · · · · · · · · · · · · · · · · · ·	-100-	The second secon	IFORMAT	 • ENCODE •		 OUTPUT HEADER	ACCESS-JUT	が 20mm 10mm 10mm 10mm 10mm 10mm 10mm 10mm	 · · · · · · · · · · · · · · · · · · ·	4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	100	IFORMAT	- CANERIA	a a second state of	+3000N3	THE S. P. P. P. P. P. SHELLEN SPECIAL CONT.	
																											2000	

6 S DAAPER LAGGRATORY JOYLAL STRUCTURED DESIGN DIAGRANMER INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

 ā.	 	IFORNAT	CNVERT*	*3000W3	 • TUO	-CAT	CMVEAT	120000000000000000000000000000000000000	CNVERT	IFORMAT	· · · · · · · · · · · · · · · · · · ·	END THE RESIDENCE OF THE PARTY	WATER THE PARTY OF	· · · · · · · · · · · · · · · · · · ·	out-	· · · SPACES	OUTPUT TITLE PAGE	ACCESS OUT	 · · · · · · · · · · · · · · · · · · ·	•0uT•	IFORMAT	CANEBIA	
																						Carlot a F.F.	

E S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRANNER Envocation Diagram of the Design Diagram Generator (DDG)

|--|

6 S DRAPER LABORATORY JOYTAL STRUCTURED DESIGN DIAGRAHHER INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

CHVERT		0uT•		IFORMAT	10 0 0 0 M	CNVERT*		ENCODE+	THE RESIDENCE OF THE PARTY OF T	HAX		001*	AMSE 60	KANSTER HRLIES	CAT	- Carrios	CNVERT	0 10 0 10 0 10 10 10 10 10 10 10 10 10 1	3	IF ORMAT	- 一般のないないです。	CNVERT		ENCODE+	-nax	E-Contract of the Contract of	-100-	POATE FILES	CESS3	大学 のうちょうかい 大学 のから かんしゅう	-64	CNVERT*	9700-1- 0	0u1·	4 Art - DINGS # 2.5	-IF OKHAT	CHVEPT	Ben-18 Torresponde 188 188 and 1	ENCODE+		-HAX	-01174	4990543	REMQUO+	TRANSPER BOTTER	TARMOTER HABIES
	•	•	•	:	•	•	•	•	•		•	:			:	•	• •			:	•				:			1	34-		٠.			•						•	٠.		•	:	. :	
•	•	•	•	•	•	•	•	•	•	•		• •		8														\$.		•		•	•	•			•	•	•	•			•		٠.	
•	•	•	•	•	•	•	•	•	•	•	•	• •	•	•	•	•	• •	• •	•	•	•	• •	• •	• •	•	• •	• •													*						11. Walter

G S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGNAMER ENVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

ca7	· · · · · · CHVERT*	0JT-	IFORNAT	CNVERT*	ENCODE+	 经出售的存储 三十年	01	BOX*MAP	140-	*CNVERT*	· · · · · · · · · · · · · · · · · · ·	1000年の日本の大学の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の	CLOSE'REC	ACCESS3	ca1	を2世紀2日本ので 1 · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	• T00	IFORMAT		* * * * * * * * * * * * * * * * * * *	• • • • • • • • • • • • • • • • • • •	×VIII	* * * * * * * * * * * * * * * * * * *		RENGUO+	*****	TRANSFER WRITES	· · · · · · · · · · · · · · · · · · ·	360296	CNVERT	· · · · · · · · · · · · · · · · · · ·	THE RESERVE OF THE PARTY OF THE PARTY OF	- FLORIBI
	•			•		 •		•		•		•		•		•			4 0	• •		٠		•	• •	•	•	•	•	•	•	•	ALCO PA	

E S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRAMER Envocation Diagram of the Design Diagram Generator (DDG)

CHVER TO	STORY SERVICE	EMCODE+		nax	2 - 2 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	.Ino	101 of 180 of 1	PLACE	*All Control of the C	ACCESS3	= 1 = AQ A		The Strategies	CMVER TO	在中国的 中国		*** 15.0 x 45.0	IFORMAT	** • • • • • • • • • • • • • • • • • •	· · · · · · · · · · · · · · · · · · ·	一 一 一 日 一 日 一 日 日 日 日 日 日 日 日 日 日 日 日 日		明 於國 一本十		\$ \$ \$ \$ \$ \$ \$ \$ \$	Out	28.000	RENDUD+		"TRANSFER WRITE3	の部件記載し	CAT		· · · · CNVERT*	- · · ·	• • • • • • • • • • • • • • • • • • • •		IFORNAT	· ** ** ** ** ** ** ** ** ** ** ** ** **	· · · · · · · · · · · · · · · · · · ·		+3000E+	A	XVH		a same contracts			OUT-	The second secon	IFORMAT	THE RESERVE AND THE PROPERTY OF THE PARTY OF
						*	-	!		-	-			6		*		3	1	1							5						è		100																	
		٠		210	á				9	-	•				7	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	i	•	•	•	•	•	•	•	• '		•	• •	•	•	•	•	•	•	• •	•
• •	•	•	•	•	•	•	•	•	•	•	•	•	•	, ,	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	• •	•					•	•	•	•		•
																																			1																	

G 8 DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRANMER INVOCATION DIAGRAN OF THE DESIGN BIAGRAN GENERATOR (DDG)

*CAVERT*	ENCODE+		INSERF	EXTRACT - NAME	A STATE OF THE STA	· · · · · · · · · · · · · · · · · · ·							IFORNAT		· · · · · · · · · · · · · · · · · · ·		+3000N3	The state of the s	-100	TO DE MOILU	· ·	STORES OF STREET		· · · · · · · · · · · · · · · · · · ·	* * * *	• • • • • • • •	ALL PROPERTY OF THE PROPERTY O	CNVERTO	SO SOLLIEMETH STREETS		· · · CNVERT		• ••••••	a new California	MZEKK	** ** ********************************		TRANSFER- WRITE3		car		CH WERT.	The state of the s	· · · · · · · · · · · · · · · · · · ·		IFORNAT	areausons.	
	•	•	• •	•	•	•		•	•	•	•	•	•	•	•	•	• '	•		•	•	•	•	•	•	•	• •	• •	•	•	•	•	•	•		•	•	•	•	•	•	•	•	• •	• •	٠.	• •	•
• •	•	•	• •	•	•	• •	•				•	•	•	•	•	•	• •	•				•	•	•	•	•	• •				•	•	•				•	•	•	•	•	•	•	• •		• •		•
		•											•	•											•							•										•						

AMMER DOG 3

ENCODE+		***************************************	TRANSFER METEN		· ·CHVERT		**************************************	*IFORMAT		CHVERF	FACOR		nax	***************************************	STR	CAT	CHVERT*	0011	-CNVERT*	-ENCODE +	-0uT+	ATHERS 2.0		NSFER- WRITET		CAT	CNVERT*		IFORMAT	CNVERT*	WORL STREET DESIGNATION
			!	ā.	SE W				1000		35.00			200	Seg	I.			Ī.		Ī	HZERR	5	TRA		Ĭ.			Ī.		
••				•		•		•	•			•		•								- 6.42	. !	. :							
••	• •					•		•	•			•			• • •										•						•
												•																			
••																															
									•			•																 •			

E 8 DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANNER INVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

	•		-TRANSFER WRITEL		647	· ·CNVERT*	· Contraction of the contraction	.170		- Trowns	· ·CHVERT		ENCODE+	****	*386995****	•out•		MCEST COLUMN		- Control of the cont	* ***********			1894246467471		Coldin	ANSFER-URITES		TOTAL EMB C	· CHVERT.		-1001	- TEDBHAT		*CHVERT*		ENCODE+		HAX	ouT*	o the state of	TRANSFER WRITE	-CATURE BITCHES SCHOOL B
								• •				•				•		-			#IH			PHZERR			TR		-		•				•	•			•				10 See
•	•	• •	•	•	• •	•	•	• •	• •	•	•	• •	•	•	•	•	• •	• •	•	•	•	• •	•	•	• •	•	•	• •	•	•	•	• •	•	•	•	•	•	•	• •	•	•	•	
				:	• •	•	:	•		•	:	•			:	•	• •		•	:	•	• •	•		• •		•			:	•	•						•	•				
																																	N.	The ASSESSED									ACA WELDON

6 S DRAPER LABORATORY JOYLAL STRUCTURED OESIGN DIAGRAHMER Invocation diagram of the design diagram generator (DDG) -- TRANSFER WEITES \*\*\* TRANSFER WRITE ---CNVERT ---CHVERT ---CHVERT ---CHVERT --IFORNAT --ENCODE+ --IFORMAT .--CHVERT ---CNVERT \*--OUT\* ---OUT. --- TUSH'LAYOUT - INFO -- IFORHAT ---OUT. ------------0uT-\*\*\* PHZERR \*--0UT\*

: S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANHER (WVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

RENGUO+	RENGUO+	RESOLVE.STUMP	4006883	ca7	· · · · · CNVERT	•100	IFORMAT	· · · · · · · CNVERT*	· · · · · · · · · · · · · · · · · · ·	nax	001	REMQUO	TRANSFER-WRITES	CAT	· · · · · · · · · · · · · · · · · · ·	•100	· · · IFORMAT	CNVERT*	ENCODE+	HAX	our.		ca1	· ·CAVERT	***************************************	IFORMAT	· ·CNVERT*	ENCODE+	
•	• • •	, .	••	• •	•	• •	•••	••		• • •	••	•••	• • •	• • •	• •	•,	• • •	•	• • •	• •	• •	•	• •	•	• •		•	•	The state of

? S DRAPER L'ABORATORY JOYLAL STRUCTURED DESIGN DIAGRANNER INVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

FEAI - ETIIE		our-	格式の対象を	PHZERR					· · · · · · · · · · · · · · · · · · ·		· · · · · · · · · · · · · · · · · · ·	• 1.00		TAUMAT.	· · · ·CNVERT*		ENCODE+	XVII		· ·TRANSFER HRITEL	CAT	· · · · · · · · · · · · · · · · · · ·	0.00	• 100	· ·-IFORMAT	· · · · · · · · · · · · · · · · · · ·	ENCODE+		• • • • • • • • • • • • • • • • • • • •	PLACE		ACCESS3	CAT	CNVERTE	***************************************	· · IFORMAT
٠	•	•	•	•	•	• •	•	•	•	•	•	• •	•	•	•	•	• 1												•		•		•	•		 Manager Control

C S BRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRANMER INVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

CNVERT	ENCODE+		001-	REMQUO+	TRANSFER-WRITE3	-car	CNVERT	001-	IFORNAT	CNVERT*	ENCODE+	747 ×Vu-		car	CNVERT*		FORHAT	CNVERT	ENCODE+	INSERT	EXTRACT*NAME	ACCESS2	car	CHVERT	•100	IFORMAT TANGET OF		
2.4.4	•	•••	•••	* 1		•			• •	•		8 6	•			•		•						 •		IN CONTRACTOR	S. Commercial section	•

G S GRAPER LABORATORY JOYTAL STRUCTURED DESIGN DIAGRANNER ENVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

																																									M. Visters	
ENCODE+	-out-		-45.74000		-CNVERT*		-001•	-CNVERT*		- CENSIA	CHVERT*			PHZERR		-001-	-TRANSFED - NOTTE 1		CAT	***************************************	· · · · ·	·100	The Shape at	IF ORMAT	· ·CNVERT*	A. C. C. C. C. L.	ENCODE.			011-	- 1	-TRANSFER" WRITE	CAT		CNVERT	·	See Chicken	IF ORMAT		CNVERT	ENCODE+	•
:	: :	• •		3.	:		: 	NO			:			Hd	•			•	•	• •		•	•		•	•	• •		:		•			N. dillower								
		٠.			•						•			•					•														•					•		٠.		
•	• •	••		• •	•	•	• •	•	• •	• •	•	• •		•	•	• •	•	•	•	• •	•	•	• •	• •	•	•	• •	•		•	•		•	•	• •	•	•	•	• •	• •	•	•
•	• •	• •	•	••	•	•	• •	•	•	•	•	• •	• •	•	•	• •	•	•	•	• •	•	•	•	•	•	•	• •	• •	•	•	• •	• •	•	•	• •	•	•	•	• •	CP LA CO	•	•

G S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANNER INVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

ACOM - EMP D	W. S. C. C. C.	UBSTR		CAT	•	CNVERT®		*0ur*		CNVERT*		ENCODE +.		001*		22 00 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	-0uT*	TRANSFER WRITES		· · · · CHVERT		001		IFORNAT	CNVERTO	 -EA CADE		0uT*		TRANSFER- KRITES			CHVER!	0111.		I FORMAT	•	CNVERT*	 - EN COURT	CHARLE WAS ASSESSED.	
		:		•	•	•	•	•	•	•	•	•	•	•		PHZERR	Ī										18.00			7			•	•		200	•		200		
		•	•	•						•				•	•			 			•		•		 		•	•			•	 			•						
										•																			•										25.75 MARK		
			A SECOND						-																														A P. Day	-	

6 8 BRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANNER ENVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

· · · · · · · · · · · · · · · · · · ·	· · · · SPACE S•			007*	PHZERR POTINZESHO MSTARP	OUT-	TRANSFER'URITES	CAT CADE			001-	IFORMAT			ENCODE+	のは かるための 一分 日本の 一方の 一方の 一方の 一方の 一方の 一方の 一方の 一方の 一方の 一方	 •••••••	TRANSFER WRITE	***************************************		CNVERT	001	IFORMAT		 ENCODE+	 	•100	PUSH-LAYOUT - IMFO	 TO MARKET TO ASSURE THE TAX BOX	
•	••	••	••	•••	••				• •	•	••	•	•	••	•	• •	••	•		•	• •	•	•	•	• •	•		• •	TAKES FREEDOM EAST	

G S DRAPER LABORATORY JOYLAL STRUCTURED DESIGN DIAGRANNER ENVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

TRANSFER - WRITES		CAT	CA VERT	• 100		FORMAT	CN VEKI	EMCODE+	•	HAX	• !		TOANGEGOODET		*CAT *		· ·CNVERT		 - Trokuri	· ·CNVERT*		EN CODE+			·0uT•	4	245 B 40		¥ 00			CALLE RA	•		1.0001202				The state of the s	大学 この一般など ないかっているかいか
	•	0			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		· · · · · · · · · · · · · · · · · · ·	•	*********			Charles and Charles		大田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田	•	大学の日本で ・	•	A	MA SE	•	10 2 3 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	•	•	※日本の中で			KENGOO	IT INUE . BOX	000000	ACCESS3	CAT	Agents we	CHVERT		-100	 TO CANADA	CHVERT	5 40006	Marketon Co. Co.	*
•		•				•		•		•					•					* **				* **					•	! .	•		•	•			•	• •	Con to the same	Townson or the second

6 S GRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANNER ENVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

out-	ENGUO+	TRANSFER"WRIT	 *******	CHVER TO	·100	IFORNAT	CNVERT	* - (1 - 1) I	ENGODE+	HAX	•0uT•		ACCESSA	CAT	CAPERIT	·001	IFORMAT	6 p. c C. 10 p.	CAVERT	ENCODE+	197 - 1 - 1 - 10 - 10 - 10 - 10 - 10 - 10	 ·0u1•	REMQUO+	TRANSFER-WRITE	CAT	CNVERT	•TU0	IFORMAT	CNVERTO
												•	١.									 		 		 		 	

G S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRANMER INVOCATION DIAGRAM OF THE DESIGN DIAGRAN GENERATOR (DDG)

***···	·011.	CATCRESSE.	CNVERT*	*100	IF ORMAT	CHVERT			MAX SMCSBES	00T*	REATE FILE 4 . RECS	ACCESS2	•	car	· ·CNVERT	•Tuo •	IFORMAT	· ·CMVERT*		·		RENGUD+	ACCESS4	CAT		CHVERT	.100	•	IFORMAT	· ·CHVERT	*3000N3	
	•					• •	• •	:	• •				•				• •	• •	::	•	• •			•	•	• •	•	•	• •	•	•	

 TRANSFER URITE.	· · · · · · · · · · · · · · · · · · ·	• out	IF ORHAF		ENCODE -		•100	00x*nAP	CATCHECKERS	· ·CMVERT*		IF ORMAT	CHVERT*	• • • • • • • • • • • • • • • • • • •		out-	SPACES*	SUBSTR	ear	CNVERT	•100	CNVERT*	ENCODE+
		• • •	• • •	•••	٠,•	\$ A	• ,	• • •	• • •		•	• •	,• ,•	•	•	•,•	•••	••		•	••	•	•

G S DRAFER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

CREATÉ"H"PTR"REC	CAT	· ·CAVERT	·00.T•	IFORMAT	CNVERT*	15000000	-	 **************************************	* Section 10 section 1		TRANSFER WRITE3	car	•	· ·CNVERT	00L	IF ORMAT	CNVERTO		ENCODE+	XW	-Tuo		- 64	CNVERT			IFORMAT		CHAEKI -	ENCODE+	OUT*	REATE. Y. PTR. REC
Ş	• • •	• • •	• •	• •	• •	• •	•	 	• •	•	•	• •	•	• •	•	• •	• •	•	• •	• •	• •	•	• •	•	•	• •	•	• •		• •	• •	1

6 8 DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRANNER ENVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

·ACCESS3	A MEDICAL A		Table of the same	- CANERIL		 · · IFORMAT		· · · · · · · · · · · · · · · · · · ·		*3000K3		HAX	***************************************	1		TRANSFER' WRITES	•	car		CNVERT		IFORNAT		CNVERTO	and the second of the second o	ENCODE -	· · · · · · · · · · · · · · · · · · ·		·0ur•	*4800000	i.	CNVERT	6.7	out-		IF ORNAT	CNVERT	ENCODE+		GET'F1'REC		Teemake
• •		•	•	•	•	•	•	•	•	•	•	• •	•		•	•	•		• •	• •	٠	•	•	• •			•	•	•		•	•	•	•	•	• •				I	•	The second

G S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAHMER INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

	ERIT				:RT.	-	106+		WAS LO	Sanda						2			:RT•		100		RT.	+30				R-WRITES		-CNVER T*	00T*	
-CAT	CNVE	·01	K. 18 12 h	-IFORMAT	*CNVERI	KAN	ENCODE	-00T+			2		-CNVERT*	143-54	-ENCODE +	MITTATE RECO	CCESS3	-CAT	CHVER		100-	-IFORMAT	CHVERT	ENCODE	-HAX	-001-	-REMQUO.	-TRANSFER	CAT			
• •		•		٠.	•			:		•	-100-	I FORMA!	2		Ž	E	¥.	. :				٠.					. :	. :				
	 											1	. !			3																
• •	•	•	•	• •	•	•	•	•	• •	•	•	.!	• •	•	• •		• • •	•	• •	•	••	• •	• •	•	• • •	• • •	• •	•••	• • •		• •	

6 S DRAPER LABORATORY JOYTAL STRUCTURED DESIGN DIAGRANMER INVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

· ·CHVERT	********	ENCODE+		The second secon	·0uT•	The second secon	-ACCESS+	e parties e	CAT	CHVERT-			*TENBUAT		· ·CHVEPT		ENCODE+	* サー・ビルコを発送を	*****	240	OUT•			*TRANSFER* WRITE	- 日本の事件がよ	caT	*Contractor	CNVERT		ade i	IFORNAT		· ·CNVERT	• 50.	ENCODE+	* *************************************				-BOX*MAP		ERT	9.80	-EXTRACT NAME	ACCESS2	· · · · · · · · · · · · · · · · · · ·	car	
	•	• •			•	•						•					•	•																								E.		١.				
•	•	• •	•	•	•	•	• •	•	• •	•	•	•	•	•	•	•	•	•	•	•	•	•	• •	•	•	•	•	•	•	•	•	•	•	•	•	• •	•	•	•	•	•			• •	•		•	•

G S DRAPER LABORATORY JOYLAL STRUCTURED DESIGN DIAGRANHER INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

	· CHVER TO	.100	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	IFORMAT		CNVERT*		***ENCODE*		-0117		- DE HOUSE	- KE HEOO	The County of th		CHUEBTE	The second secon	-5.1.6	-	CNVERT	William Color Bed Self S	-LENGTH		CMVERT*		our.		HZERR	- C. S. C. S. C. C.		-TPANSEED - MOTTET	-	CAT	Service States	· ·CNVERT	. 10-	*100		IFORMAT	· ·CHVERT		ENCODE+		MAX	The state of the s	OUT-	A The Control of the Control	TRANSFER" WRITE		CAT		· · CNVERT*	BURN A SELECTION OF THE SE
***************************************	• • •			•	•	•	•	•	•	•	•					•	•	•		3		7	•	•	200	3		-			•	•	•	•	•	•	• •	•	• •	•	•	•	•	•	•	•	•	•					
***************************************				•	•	•	•	•	•	•	•	•			•	•	•	•	•	•	•	•	•	•	•	•	•	• •	•	•	•	•	•	•	•	•	• •		• •	•	•	•	•	•	•	•	•	•	•	•	•	• '	•
***************************************	• •	• •	•	•	•	•	•	•	•	•	•			•		•		•	•	•	•	•	•	•	•	•	•	• •		•	•	•	•	•	•	•			• •	•	•	•	•	•	•	•	•		•	•		• •	•
	•••			•	•	•	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•	•	•	• .	• •		•	•	•	•	•	•	•	• •		• •	•	•	•	•	•	•	•	•	•	•	•		• •	

8 SOAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRANNER ENVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

			- Trowner	· · · · · · · · · · · · · · · · · · ·	Manage of the second	+=000E+		The state of the s	• • • • • • • • • • • • • • • • • • •	日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日	SUBSIT	car	a terbalanta	· · - CHVERT*			CHVERT		ENCODE+	**************************************	10000000000000000000000000000000000000	- THE ENR	001-	The Company of the Co	- IKAMSTER WALLES	car		CHVERTO	• • • • • • • • • • • • • • • • • • • •		· ·IFORMAT	a completed files		0.100	- Encode	nax	S. Landalosson	·017•	3	KANS PER" WRITE.	 a St. Constitution of the second	· ·CNVERT*		·100
***************************************																																	•											
	•••	•	•	•	•		•	•	•	• •	•	•	•	• •	•	•	•	•	•	•	• •	•	•	• •	•	•	•		•	•	•		•					•	• •		•	•	•	• •

IFORNAT	ENCODE+	ACES	2020-26	AVOUT-THEO SEES	HZERR	-00T*	TRANSFER-URTTES	CAT-C105435454	CNVER TO	**************************************	IFORNAT	CHVERT	+3000H3	 •001	-TRANSFER- WRITE	car	· · · · · CHVER T*	*********	IFORNAT	· · · · · CHVERT*	• • • • • • • • • • • • • • • • • • •	nax
		 3	out.	7-404	¥.	٠.	١.	••	• •	• •	•	••	• •	 •	1			X	1. s.			- boton

6 S BRAPER LABORATORY JOYLAL STRUCTURED DESIGN DIAGRANNER INVOCATION DIAGRAN OF THE DESIGN DIAGRAN GENERATOR (DDG)

-POP- PROC'STACK			CHVERT	***** ***** *	IFORNAT		CHVERT	ENCODE.	The second secon	 	•100	TRANSFER" WRITES			CHVERT		IF ORMAT	· · · · · · · · · · · · · · · · · · ·	•	+3000E+	MXX	P114		TRANSFER- HR ITE	CAT		CHVER T			IFORMAT	· · · · CNVERTO		ENCODE	X111	•0uT•
* 1	•	•	•	•		•	• •	•	•	• •	•	• •	•	• •	•	•	• •	•	•	• •	•	• •	•	••	•	•	•	•	•	• •	•	• •	•	• •	•

E S DRAPER LABORATORY JOYIAL STRUCTURED DESIGN DIAGRAMER INVOCATION DIAGRAM OF THE DESIGN DIAGRAM GENERATOR (DDG)

TRANSFER-URIT	CAT	*TU0	IFORHAT	CHVER T	ENCODE	 -100	**************************************	CAT	CNVERT	•100	IFORNAT	CHVERT	ENCODE	 -100	USH PROC'STACK	CAL	CNVERT*	•T00	IF ORMAT	CNVERT*
	TRANSFER' WRIT	TRANSFER-URI		TRAMSFER-URI		 								 						

ES BRAPER LABORATORY JOYLAL STRUCTURED DESIGN DIAGRANNER ENVOCATION DIAGRAN OF THE DESIGN DIAGRAM GENERATOR (DDG)

	-PHZERR	007-	TRANSFER-WRITES	ar	· · · · · · · CHVER T*	OUT-	IFORMAT	· · · · · · · · · · · · · · · · · · ·	ENCODE+	x	-100	TRANSFER-HRITE.		CNVERT*	•001	IFORMAT	· · · · CNVERT	ENCODE+	 •Tu0
# B					2 6	* *					4 4		5 4	4 4 4					

CONTINUATIONS AND INDEPENDENT ROUTINES

## MISSION of Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C<sup>3</sup>) activities, and in the C<sup>3</sup> areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microusve physics and electronic reliability, maintainability and compatibility.

an concentration concentration contented to

